**Application éducative**

Technique

# Installation

1. Récupérer le projet : soit en copiant/collant, soit avec git (git clone…)
2. On installe les dépendances du projet. Avec composer, php composer.phar install
3. Créer une base de données
4. Renseigner les informations concernant la base de données dans le fichier app/config/parameters.yml
5. Générer l’ensemble des tables de la base :   
   php bin/console doctrine:schema:update --dump-sql
6. L’application est maintenant opérationnelle

# Mise à jour

## Mise à jour du framework symfony

Il peut avoir des mises à jour de symfony en terme de sécurité et/ou de performance.

Php composer.phar update

## Mise à jour du projet sur git

Mettre en ligne des modifications effectuées par les développeurs de l’application

Git pull origin master

# Architecture de l’application

## Identification

Toutes les pages nécessitent d’être authentifiés pour pouvoir assurer un suivi individuel et sécuriser l’espace « enseignant ».

L’identification s’effectue avec « **prenom.nom** ». Il peut également être personnalisé pour certains cas depuis la fiche d’un utilisateur.

Le mot de passe est facultatif mais recommandé pour les enseignants.

A la première utilisation, un compte est créé par défaut pour assurer la première connexion. Le compte peut ensuite être supprimé si besoin.

Identifiant : **root**

Mot de passe : **projetmasterwic2017**

## Technique

### Fichiers de configuration

**App/config/parameters.yml**

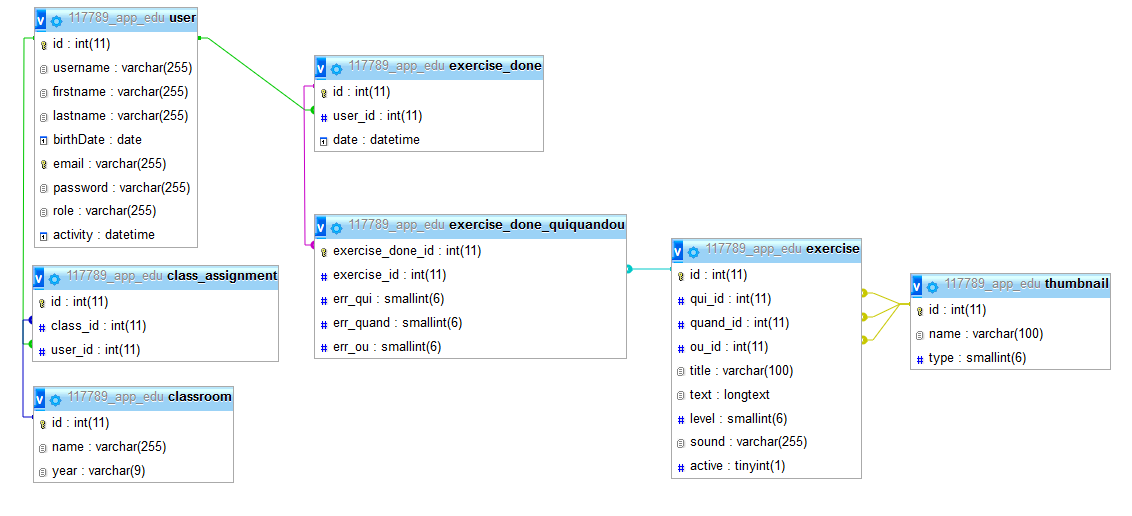
* Configuration de la base de données

**App/config/config.yml**

* Api\_voice\_key : Clé pour la génération vocale automatique des textes
* Root\_password : Mot de passe du compte root créé automatiquement

### Base de données

* Classroom : cette table contient la liste de l’ensemble des classes existantes.
* Class\_assignment : on affecte ici à un élève, une classe en particulière.
* Exercise : les différentes histoires avec en plus les réponses correctes.
* Exercise\_done : exercices effectués par les élèves.
* Exercise\_done\_quiquandou : exercices effectués par les élèves avec des informations complémentaires pour l’atelier « qui ? quand ? où ? ».
* Thumbnail : ensemble des vignettes utilisées pour les histoires.
* User : Utilisateurs utilisant l’application (élèves et enseignants).



## Organisation du projet (symfony)

### Les répertoires

Un projet symfony contient toujours les mêmes dossiers à la racine du projet.

* App : contient les éléments/paramètres généraux de l’application.
* Bin : des exécutables pour pouvoir réaliser certaine action automatiquement via des commandes dans une console.
* Src : le dossier dans lequel nous travaillons le plus. Il contient lui-même chacun de nos bundles et c’est donc ici qu’on retrouve la majorité de notre code.
* Tests : répertoire lié à tout ce qui concerne les tests.
* Var : symfony écrit dans ce répertoire tout un tas d’informations comme des logs, du cache, etc.
* Vendor : contient toutes les bibliothèques externes à notre application.
* Web : on retrouve dans ce répertoire les éléments dont les utilisateurs du site ont besoin comme les images, fichiers css, javascripts… Il s’agit d’ailleurs du seul répertoire accessible par ceux qui visitent le site.

### Bundles

Un bundle est une « brique » de notre application. Cela permet de regroupe des fonctionnalités de même nature. Dans notre projet, on en dénombre trois :

* AppBundle : il s’agit d’un bundle général permettant de faire le lien entre les autres.
* ExerciseBundle : on retrouve dans celui-ci tout ce qui concerne l’atelier « qui ? quand ? où ? ».
* SchoolBundle : dans celui-ci, on a regroupé tout ce qui est lié à une école. On compte notamment les élèves, enseignants, classes, etc.