

# Oenoweb



---

## Projet Oenoweb

Préparé pour : David Janiszek  
Drouhin Antoine - Mahmoudi Maxime  
Avril 2015  
Groupe : L2AB1



---

**WEB SERVICE**

## INSTALLATION

### Prérequis

La machine d'accueil devra avoir préalablement installé :

- MySQL
- Git
- PHP 5.0
- Composer

Nous vous conseillons vivement de posséder des versions à jour de ces logiciels.

### Mise en place

Etape 1 : Copier le script d'installation dans le répertoire ou vous souhaitez installer le projet

Etape 2 : Exécuter le script d'installation

Etape 3 : Remplissez les informations relative à l'accès a votre base de données MySQL

---

## WEB SERVICE

# DOCUMENTATION - OENOWEB

## Objectifs

Le but de cette partie du projet était de fournir un accès à une base de donnée de vins pour l'application Oenologik, ainsi qu'une page web de présentation du projet.

## Technologie

Nous avons choisi de développer ces services en php, et d'utiliser une base de donnée en SQL. Nous avons également fais le choix du framework Symfony2 qui nous à permis d'organiser notre travail proprement.

## Relation d'utilisation avec d'autres modules

La relation entre cette partie du projet et l'application se fait au travers d'un web service développé selon le formalisme REST, le fonctionnement du web service sera détaillé plus loin.

## Organisation des répertoires

L'organisation des répertoires ce fait selon une organisation Symfony classique :

- **A la racine :**
  - Le dossier « **Documentation** » contient l'ensemble des documents qui illustre le fonctionnement du projet
  - Le dossier « **BDD** » Contient les différentes sauvegarde de la base de donnée MySQL qui ont pu être faites.
  - Les dossiers « **bin** », « **vendor** » et « **web** » Contiennent l'ensemble des ressources de Symfony relatives au projet, nous n'avons pas modifié manuellement leur contenu.
  - Le fichiers **composer.json** comporte la liste des bundles installé dans le projet.
  - Le fichier **exitcache.sh** est un script BASH (unix) permettant de vider le cache et de redéfinir les droit qui y sont relatif automatiquement. (Le cache est une source de bug dans symfony et automatiser cette action nous a fais gagner beaucoup de temps.

- Le dossier « **app** » contient notamment :
  - Le « **cache** » de l'application
  - Le répertoire « **config** » qui permet de paramétrer l'application au travers de plusieurs fichier de configuration.
- Le répertoire « **src** » contient le coeur du projet à savoir :
  - Le **Bundle Oenoweb**, qui centralise les différentes fonctionnalités que nous avons crée dans lequel on retrouve :
    - Le répertoire « **Controller** », qui contient les contrôleurs rest c'est à dire l'essentiel de notre travail, ce sont la définition du comportement des réponses XML ou JSON en fonction de paramètres.
    - Le répertoire « **Entity** » qui contient l'ensemble des tables de la base de donnée en DQL (Doctrine Query Language). Ecrite en php. Le répertoire « Repository » y est associé mais nous ne l'avons que peu modifié.
    - « **Form** » Contient les formulaires utilisé pour les routes POST du bundle REST
    - « **Ressources** », contient l'ensemble des fichiers de configuration relatifs a OenowebBundle.

## Procédures externes : Liste des routes du Bundle REST

### Routes GET

Les routes GET renverront au choix un fichier xml ou Json contenant les données demandées. le format de retour devra être spécifié dans le paramètre `{format}`. Si ce paramètre n'est pas remplis alors le format de retour par défaut sera le xml.

#### La liste des régions :

route : `ADRESSE_SERVEUR/api/regions.{_format}`

#### La liste des AOC par régions :

`ADRESSE_SERVEUR/api/regions/{region}.{_format}`

#### La liste des domaines par AOC et Régions

route : `ADRESSE_SERVEUR/api/regions/Provence/aocs/{domaine}.xml`

**Obtenir les informations sur un Vin :**

Exemple de XML : route : ADRESSE\_SERVEUR/api/regions/{Parametre\_region}/aocs/{Parametre\_AOC}/domaines/{Parametre\_domaine}.xml

**Afficher les Avis relatifs à un vin :**

route : /api/lists/{idVin}/commentaire.{\_format}

**Afficher les Avis relatifs à un utilisateur :**

route : ADRESSE\_SERVEUR/api/commentaires/{idUser}/user.{\_format}

**Obtenir les informations sur un utilisateur :**

route : ADRESSE\_SERVEUR/api/users/{username}.{\_format}

**Route de synchronisation de l'ensemble de la base de donnée de vins :**

route : ADRESSE\_SERVEUR/api/synchro.{\_format}

**Routes Post****Ajouter un Favoris :**

route : ADRESSE\_SERVEUR/api/favoris.{\_format}

**Ajouter un Avis :**

route : ADRESSE\_SERVEUR/api/avis.{\_format}