**GENEAGER CMS**

# 1 – Généralités

## 1.1 - Introduction

Gestionnaire de contenu GENEAGER est destiné pour la généalogie et ayant pour but de démocratiser d’avantage l’indépendance numérique dans le monde de la généalogie: offrant ainsi donc entre-autres un meilleur contrôle des données saisies (la révocation par exemple).

## 1.2 – Licence et droits

L’ensemble des codes du projet sont sous licence Creative Commons 4.0 (CC BY-NC-SA 4.0), cela signifie que :

**- Vous pouvez partager** (copier et distribuer et communiquer le matériel par tous moyens et sous tous formats)

- **vous pouvez adapter:** vous devez distribuer vos contributions sous la même licence que l'original.

- **Vous ne pouvez pas** utiliser les réalisations à des fins commerciales

Plus d’informations sur la licence sur la page suivante: <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

# 2 – Structure des fichiers

## 2.1 – Généralités

L’architecture du CMS est de type MCV.

Dossier « root » (contient le .htaccess)

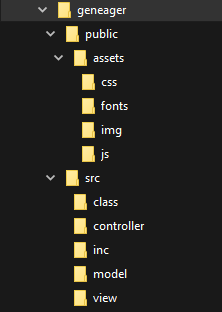
Accessible par le client (via l’URL)

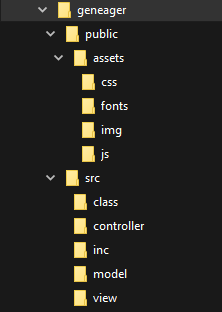
Inaccessible via le terminal client (réécriture à l’aide du fichier .htaccess).

Si le visiteur tape: *https://monsite.tld/class/index.class.php*

Il accède en réalité au fichier: */public/class/index.class.php*

🡪 En occurrence ici, il aura une page « 404 ».





**Contient les dossiers:**

- CSS (feuilles de style)

- Fonts (Police d’écriture)

- Img (icônes, sprites, …)

- JS (Javascript)

**Contient:**

- class (class appelées manuellement (sans autoloader))

- controller (programmes de logiques)

- inc (les fichiers includes (fonctions))

- model (class chargées automatiquement (autoloader))

- view (contient principalement le code HTML (la vue)): appelé avec l’autoloader si le « controller » est absent.

## 2.2 – L’autoloader

L’autoloader est une fonctionnalité qui permet de charger les bons fichiers en fonction de l’URL.

Par exemple « *https://monsite.tld/home* » appelera les fichiers suivants :

- /src/model/home.model.php

- /src/view/home.view.php (si le « controller » est absent)

- /src/controller/home.controller.php (programme logique devant appeler la « view »)

Les fichiers « models » et « views » ainsi que « controllers » sont liés ensemble notamment à l’aide du « router PHP ».

Toutes les classes générales (utilisées dans plusieurs « controllers ») devront-être rangés dans le dossier dédié nommé « class » (situé dans le dossier « src »).

# 3 – Convention de nommage

## 3.1 - Généralités

Nous utiliserons l’**anglais** pour l’ensemble des nommages.

Nous commencerons chaque nommage tel qu’il soit par une minuscule. Le 2ème mot (particule) commencera avec une majuscule et sans espace ni « underscore » (tiret bas).

*Exemples:*

*- maClassIdentite*

*- monFichier.php*

## 3.2 – Models, views et controllers

Chaque fichier MCV devra être rangé dans son dossier attitré avec la bonne double extension.

*Exemples:*

*- /model/model/home.model.php*

*- /src/view/home.view.php*

*- /src/controller/home.controller.php*