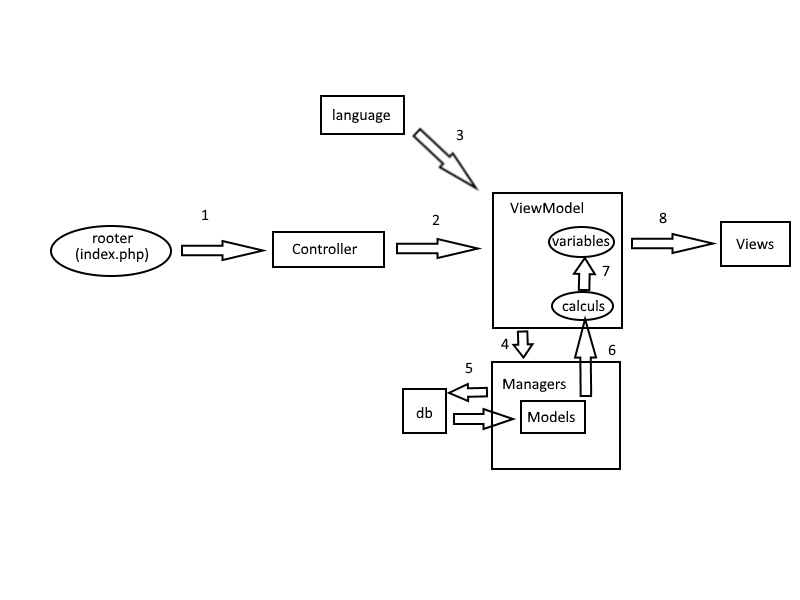
Explication simplifiée de l’architecture logicielle MVC-VM :

*Un utilisateur demande une page depuis son navigateur. Un Rooter va appeler, parmi ses Controllers, le Controller particulier pouvant répondre à la requête. Le Controller va instancier un ViewModel qui aura pour charge de récupérer les informations de la base de données, effectuer les opérations dessus, et envoyer le résultat des opérations à la vue pour affichage.*

*Son principal intérêt par rapport à un MVC classique est qu’il permet de soulager le Controller en déléguant tous les calculs sur les Models au ViewModel.*



1 : On va aller chercher la fonction à executer dans les controllers à partir du paramètre action dans index.php.

2 : La fonction du controller instancie un ViewModel particulier.

3 : Le fichier de langue est implémenté à partir du paramètre lang de l’index.php

4 : Le ViewModel instancie des Managers pour aller chercher les informations dont il a besoin.

5 : Le Manager récupère les données enregistrées dans la base de données et instancie les Models pour les stocker et faciliter leur utilisation future.

6 : Le ViewModel effectue des opérations à partir des données renvoyées des Managers.

7 : Le résultat de ces opérations est stocké dans des variables pour l’affichage dans la vue.

8 : On envoie la vue à l’utilisateur et on se sert des variables créées pour l’affichage.

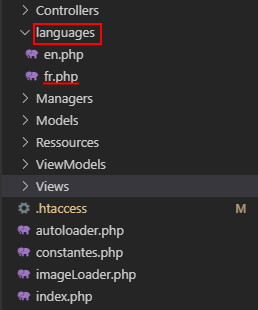
Gestion des langues :

Le rooter va lire le paramètre langue passé dans l’URL :

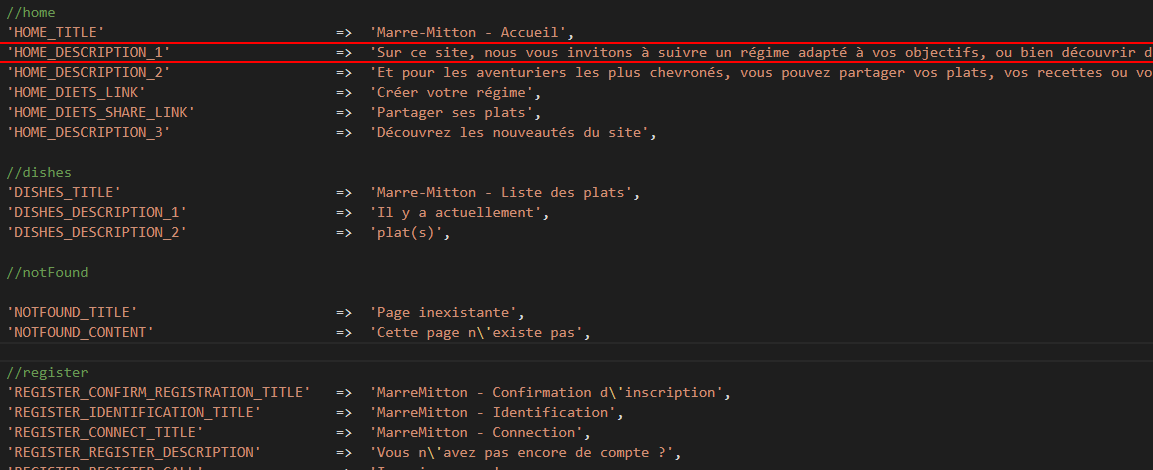
Ex : <http://marre-mitton.fr/home.fr>

La partie grise représente le nom de domaine du site, la partie rouge le paramètre action et la partie verte, le paramètre langue.

On appelle le contrôleur particulier gérant l’action en paramètre. Ce contrôleur va lui-même instancier un ViewModel à qui il va déléguer la logique de l’application. Ce ViewModel va chercher dans le répertoire lang/, le fichier de langue composé d’un tableau contenant toutes les strings de l’application et le « charger ».



*Position des fichiers de langage dans l’architecture*



*Une partie du contenu de fr.php*

Il importera ensuite les strings nécessaires dans la View :

