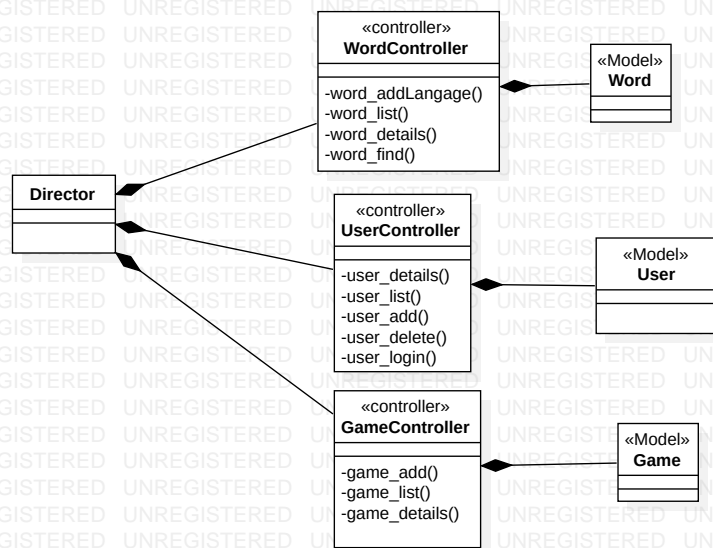
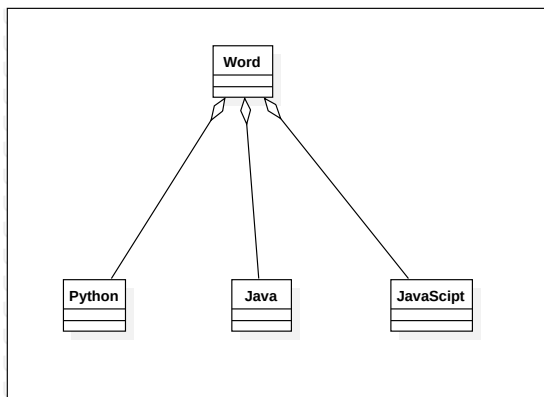


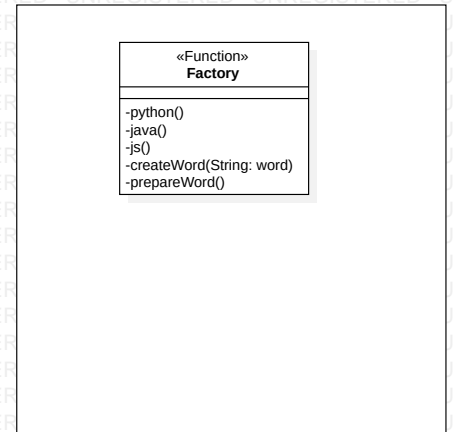
Le premier patron de conception est la façade. Faire une application web demande plusieurs liens interconnectés et peut rapidement devenir complexe. Dans ce contexte, cela permettrait d'avoir un fichier qui redistribue les demandes vers les autres fichiers. Si demain une nouvelle technologie arrive, nous aurions simplement à changer le contenu de la méthode sans à modifier les appels à travers tous les fichiers du serveur. Il va être aussi plus facile d'ajouter de nouvelles fonctionnalités car nous avons simplement à ajouter une ligne dans le contrôleur et il va s'occuper de l'apporter au contrôleur en question.



Mon site web pourrait bénéficier d'utiliser le patron de conception "Factory". À plusieurs reprises, je vais devoir construire des mots. Le fait de centraliser leur création permet d'avoir une certaine flexibilité au niveau de leur diversification. Au moment de la création d'un mot, j'aurais simplement à faire "factory.createWord("python",10)". Le premier argument est le langage qu'on veut et le deuxième est le nombre de mot qu'on veut avoir. De plus, nous allons pouvoir faire la préparation des mots à l'intérieur même de la fonction.



Plan original



Façon factory

sd Communication interne

:Page html

Directeur

Contrôleur

1 : Click sur un lien

2 : fetch

3 : Redirige vers le bon contrôleur

4 : Effectue la demande

5 : Affichage des nouvelles informations

6 : Retourne les informations demandées

Dans ce contexte, le directeur est le fichier JS qui va s'occuper de recevoir toutes les demandes et les redistribuer aux bons contrôleurs

sd DS_CreationMot

: Game

: Factory

: Director

: WordController

: NounController

1 : init

2 : CreateWord(string,int)

loop

[n < words]

3 : fetch(string)

4 : word_find()

5 :

6 : prepareWord()

7 : fetch(string)

8 : noun_find()

9 :

10 : return words

