

# Cartographie de portails Wikipédia

## Compte rendu de la réunion du 18 février

### Choix techniques

-Parser JS côté client

JS se connecte à un proxy PHP, va chercher sur wikipédia, analyse la page wikipédia, renvoie les liens au serveur. Si la page est intéressante, on remplit la table associative (cf modèle de la BDD). L'éditeur peut forcer la prise en compte d'une page.

-Utiliser greasemonkey pour éviter les problèmes de cross-domaines ? Si l'administrateur travail sur sa page en local, il n'y a pas de problèmes de cross-domaines.

-Faire des tests de monitoring pour voir s'il y a besoin d'utiliser le proxy.

-Voir comment passer en mode robot pour pouvoir effectuer plus de requêtes à la fois (50 requêtes en utilisateur lambda, 500 en robot)

-Développement de la BDD avec mysql et du PHP orienté objet

### Use Case Diagram

#### Changement:

-Un utilisateur anonyme doit pouvoir personnaliser une carte, mais seul un utilisateur authentifié peut enregistrer sa carte.

- Modérateur devient éditeur (responsable du contenu)

### Base de données

-Ajouter le nombre de modifications d'une page

-Importance d'une page : réfléchir à comment la calculer

- ! Pas de datetime négative

-Stocker les pages refusées ?

-Rajouter les données relatives à l'authentification de l'utilisateur

### Exemple de test unitaire

Un lien regroupant tous les tests associés à une fonction, mais annonce claire des résultats :

Tous tests validés : oui/non

test 1 validé oui + rapide description de ce qui est validé

test 2 validé oui + rapide description

puis détails en bas

### Pour le 11 mars

-Début/test du viewer (prototype de base) et de la timemap

-Modèle de données des spécifications de la carte (source de données, date, titre..) et de la légende (personnalisation).

Pour la personnalisation:

1. durée d'affichage de la bataille ...

2. taille de l'objet

3. symbole, couleur éventuellement