

Spécification Wikitty publication module de synchronisation

Manoël FORTUN

1^{er} avril 2011

1 Définitions

1.1 Objets wikitty <-> Fichier

Un fichier est défini par un nom, une extension, un contenu et un chemin. dans wikitty publication les fichiers sont convertis en fonction de leurs type en objet wikitty. les fichiers sources sont convertit en WikittyPubText et les fichiers binaires(eg image, etc) en WikittyPubData.

Les deux types d'objet ont les mêmes attribut :

- Name : correspondant au nom du fichier
- MimeType : correspondant au type, qui donnera l'extension
- Content : le contenu binaire pour les PubData et textuel pour les PubText

Le mimetype donne l'extension par exemple pour un PubData si on a en mimetype "image/png" alors l'extension du fichier associé sera ".png". Dans le cas d'un PubText le mimetype donnera l'extension et le langage de la source par exemple si on a en mimetype "application/javascript" le langage est javascript et l'extension sera donc ".js".

A ces objets wikitty on associe un wikittyLabel, c'est un objet qui peut contenir un ensemble de label différent, un label par exemple "src.org.chorem.entities" sert ici pour contenir le chemin menant au fichier sur le file system. Un wikitty peut avoir un certain nombre de label, pour les wikittyPub cela indiquera qu'ils appartiennent à plusieurs arborescence.

1.2 Propriétés

1.2.1 WikittyService cible

L'adresse du wikitty service auprès duquel notre espace de travail(ensemble des fichiers provenant d'un wikitty service) est lié sera stocké à la racine du dis espace de travail(premier dossier), dans un dossier caché ".wp". Cette adresse sera dans un fichier de propriété très certainement.

1.2.2 Versions des wikitty et labels

Tout objet wikitty dispose d'une version qui est modifier par le wikitty service à chaque modification de l'objet wikitty, les wikittypub donc aussi. Cette information de la version sera stocké dans un fichier de propriété dans le dossier caché ".wp" afin que l'on garde trace des versions des objets que l'on aura transformé en fichier, cela pour mieux gérer les commit et update.

On conservera trace ainsi dans ce même fichier de propriété des labels, cela permettra de pouvoir comparer les différences entre ce qui existe sur le file system et les objets présents dans le wikitty service, et donc de voir les suppressions, créations, etc.

2 Fonctionnalités

2.1 Import

-> En tant qu'utilisateur de wikitty-publication je dois pouvoir importer auprès d'un service wikitty un ensemble cohérent (eg : une arborescence) de fichier. Si le fichier est un fichier source il sera stocké en tant que wikittypubtext avec en nom son nom, un label qui sera le chemin du fichier et son extension sera convertit en type mime attribut de l'objet. Si le fichier est un fichier "binaire" (eg image) il sera stocké en tant que wikittypubdata, et son nom et label seront déterminé comme pour une source. On peut importer un fichier récursivement si on le souhaite.

2.1.1 Définitions

```
* ''import [--recursion (true|false)] [url du WikittyService] [directory]
```

2.1.2 Prototype commande et exemples

2.2 CheckOut

2.2.1 Définitions

-> En tant qu'utilisateur de wikitty-publication je dois pouvoir checkout auprès d'un service wikitty en donnant un label, je récupérerais un fichier dont la localisation dépendra du label, sous un nom dépendant de son name et une extension en fonction du mimetype. Si l'opération checkout est demandé récursivement je récupérerais l'ensemble des fichiers stocker sous le même label.

2.2.2 Prototype commande et exemples

```
* ''checkout [--recursion (true|false)] [url du WikittyService] [Label à extraire] [directory local d'accueil]''
```

2.3 Relocate

2.3.1 Définitions

->En tant qu'utilisateur de wikitty-publication je dois pouvoir changer l'adresse du service wikitty cible de mon espace de travail. Cette opération ne pourra être exécuté que à partir du fichier racine de l'espace de travail, donc celui possédant le fichier caché contenant l'adresse du wikitty service. Celà implique que l'on enregistre dans le fichier caché de propriétés que l'on effectuée cette action de relocate, et que du coup commit et update ne se passeront p'etre pas de la même façon.

2.3.2 Prototype commande et exemples

```
* ''relocate [nouvelle url du WikittyService par défaut] [directory a  
relocaliser]''
```

2.4 Commit

2.4.1 Définitions

->En tant qu'utilisateur de wikitty-publication je dois pouvoir commit auprès un service wikitty un ensemble cohérent(eg : une arborescence) de fichier sous un même label, si un tel label existe déjà sur le wiki on regardera la version, si la version du service est antérieur ou équivalente, alors la précédente version sera écrasé par notre version. Le commit commitera aussi les fichiers qui n'existait pas avant et fera les delete conséquent

2.4.2 Prototype commande et exemples

```
* ''commit [--recursion (true|false)] [--ws (url du WikittyService)] [répertoire  
à pousser]''
```

2.5 Delete

2.5.1 Définitions

->En tant qu'utilisateur de wikitty-publication je dois pouvoir delete un fichier ou dossier de mon espace de travail. Concrètement il faudra supprimer le label associé au wikitty, et si le wikitty se retrouve sans label, alors il devra être supprimé du wikitty service.

2.5.2 Prototype commande et exemples

```
* ''delete [--ws (url du WikittyService)] [répertoire ou fichier à supprimer]''
```

2.6 Update

2.6.1 Définitions

->En tant qu'utilisateur de wikitty-publication je dois pouvoir update mon espace de travail, pour les fichiers sources on effectue un diff pour mettre en relief les différences entre local et distant, pour les fichiers binaire on écrase, pour les fichiers qui localement ont été supprimé on ne les remplacent pas. Pour les fichiers qui ont été supprimé sur le serveur, on les supprime localement aussi.

2.6.2 Prototype commande et exemples

```
* ''update [--recursion (true|false)] [--ws (url du WikittyService)] [répertoire  
à mettre à jour]''
```