

# Cloud

## **Ressources :**

- [Phalcon API](#)
- [Phalcon-jQuery API](#)
- [Phalcon-jQuery exemples](#)

## **-- Contexte**

Vous travaillez pour une entreprise qui souhaite mettre en place un système de Storage as a service (STaaS), permettant de mettre à disposition de ses clients un stockage de fichiers à distance. Vous êtes en charge d'une partie de l'application web permettant de gérer ce service.

## **-- Règles de gestion**

Les clients peuvent disposer d'espaces de stockage (nommés disques) permettant de stocker leurs données. Chaque disque est loué à un certain tarif par le client ; le tarif comprend :

- Un prix (mensuel)
- Un quota (capacité de stockage exprimée dans une unité (o, Ko, Mo, Go ou To)
- Une marge de dépassement, à ne pas dépasser sous peine d'être facturé d'un surcoût.

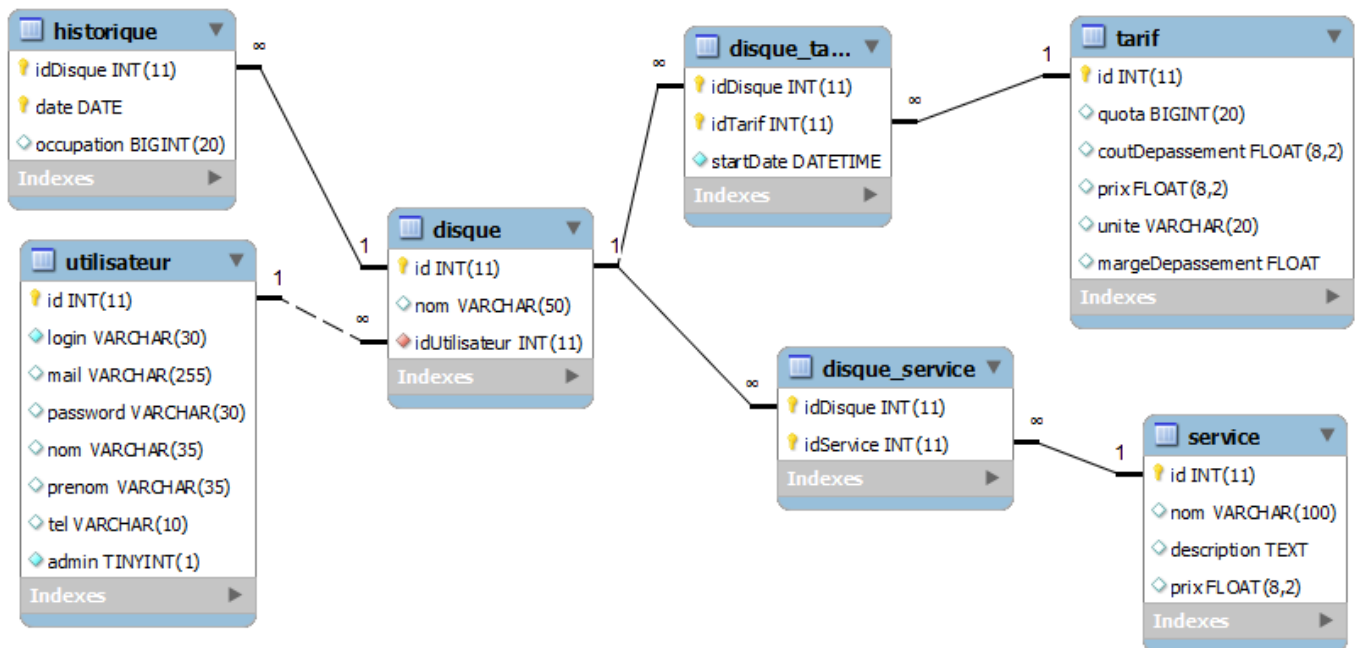
Le client a la possibilité de changer de tarif à tout moment pour chaque disque, de façon à diminuer ou augmenter son quota, en fonction de ses besoins. L'historique des tarifs associés à un disque est conservé.

Le tarif courant appliqué à un disque est celui correspondant à la date la plus récente.

Chaque location de disque peut être associée à des services (backup, loadBalancer...).

L'historique d'occupation des disques est obtenu de manière automatisée par un script PHP lancé de manière quotidienne par une tâche CRON.

## **-- Schéma de la base**



[Script de création de la base Cloud à exécuter sur votre serveur](#)

## -- Fonctionnalités à implémenter

- Créer la base de données [Script de création de la base Cloud à exécuter sur votre serveur](#)
- Faire un fork du projet <https://github.com/jcheron/phalcon-cloud>
- Cloner votre copie du projet en local pour travailler
- Faire ensuite des Commits/Push réguliers

## -- Récupération du tarif actuel (2 points)

### //TODO 4.1

Dans la table **Disque\_tarif**, sont stockés les tarifs appliqués à un disque à partir d'une certaine date (champ **startDate**). Le tarif actuel d'un disque correspond donc au dernier tarif appliqué, celui dont la startDate est la plus récente.

En utilisant les relations définies sur les modèles, et en privilégiant une approche objet, implémentez la méthode retournant le tarif actuel d'un disque dans la classe **ModelUtils** :

```

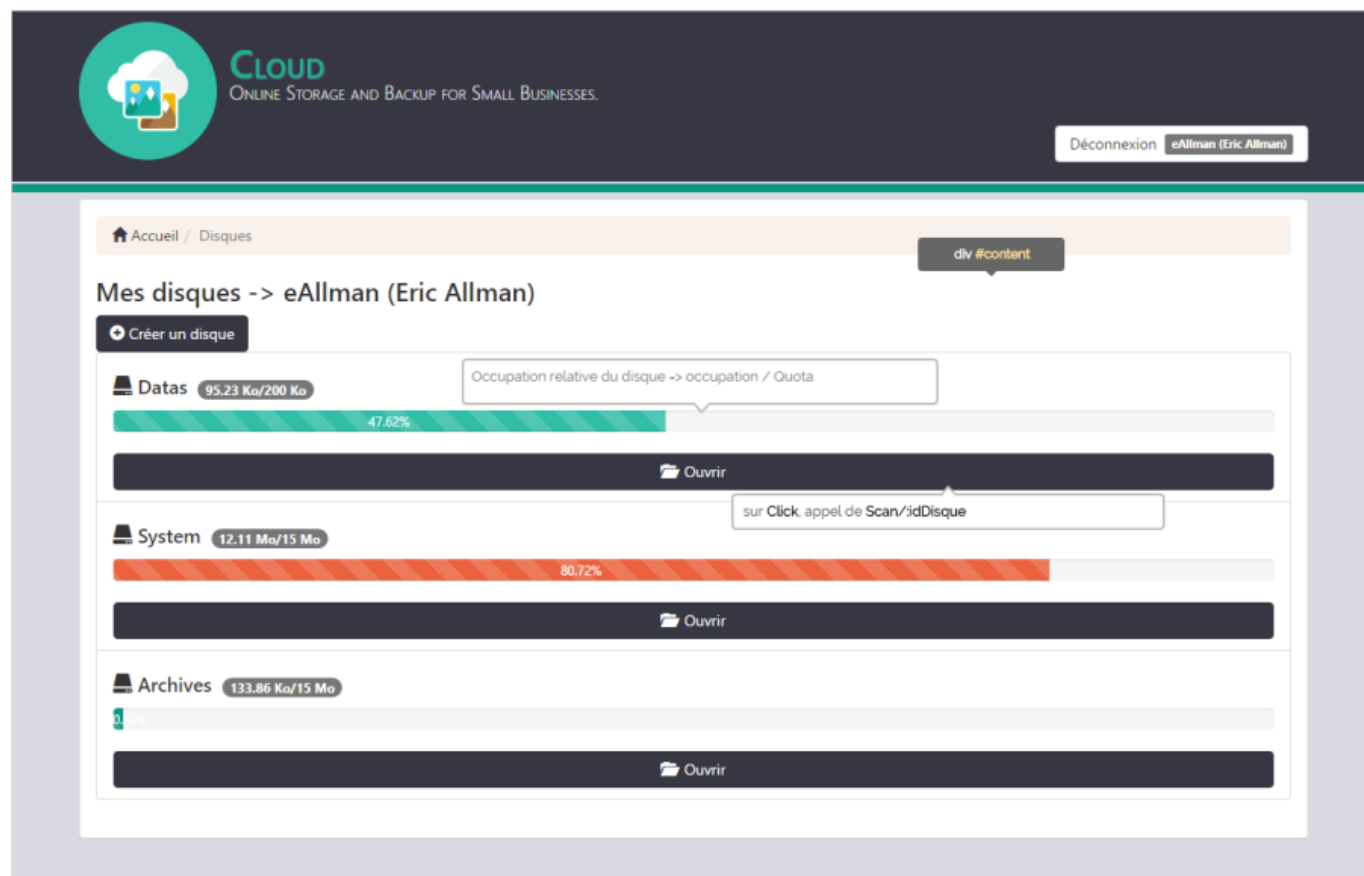
/**
 * Retourne le tarif appliqué actuellement à $disque
 * @param Disque $disque
 * @return Tarif tarif actuel de $disque
 */
public static function getDisqueTarif($disque){
    //TODO
}
  
```

## -- Affichage des disques d'un client (7 points)

### //TODO 4.2

A l'adresse **MyDisques/index**, on souhaite afficher les disques de l'utilisateur actuellement connecté.

L'utilisateur connecté est obtenu par l'appel de la méthode **Auth::getUser(\$controller)** où **\$controller** est le contrôleur actif.



### Indications

Elément	Indications
Contrôleur	MyDisques
Action	index
Utilisateur connecté	L'utilisateur connecté est obtenu par l'appel de la méthode <b>Auth::getUser(\$controller)</b> où <b>\$controller</b> est le contrôleur actif.
Occupation, Quota	le quota est obtenu sur le tarif actuel du disque, la classe <b>ModelUtils</b> permet de connaître l'occupation en cours du disque
Composants visuels Bootstrap	on pourra utiliser les composants Phalcon-Jquery <b>htmlProgressbar</b> , <b>htmlGlyphButton</b> , <b>htmlListgroup</b>
Accès à la config du cloud	Les disques clients sont localisés dans le dossier <b>public/files</b> de l'application, et les disques y sont localisés sous le nom : <b>\$cloud-&gt;prefix-[disque.nom]</b> . Cette configuration est définie dans le fichier de <b>config.php</b> dans la variable <b>cloud</b> , accessible depuis les contrôleurs par <b>\$this-&gt;config-&gt;cloud</b>
Tailles en octets	Vous pourrez utiliser la méthode <b>sizeConverter</b> de <b>ModelUtils</b>

Le style des progressbars doit donner une indication sur le taux d'occupation en % :

Style	Valeurs
info	de 0 à 10%
success	de 10 à 50%
warning	de 50 à 80%
danger	plus de 80%

Le composant **HtmlProgressbar** dispose d'une méthode **setStyleLimits** permettant de modifier le style de la **progressbar** en fonction de sa valeur

## -- Affichage d'un disque (6 points)

### //TODO 4.3

Apportez les modifications aux relations du modèle **Disque** pour pouvoir accéder à ses services (ManyToMany ?).

A l'adresse **Scan:/idDisque**, afficher l'interface suivante :

The screenshot shows a web interface for 'Cloud ONLINE STORAGE AND BACKUP FOR SMALL BUSINESSES'. The user 'eAllman (Eric Allman)' is logged in. The breadcrumb trail is 'Accueil / Scan / index / 2'. The main section is titled 'System' and has tabs for 'Home', 'Listing', and 'Upload'. The 'Home' tab is active, showing 'Caractéristiques du disque'. The details include:
 

- Nom**: System (with a 'Modifier...' button)
- Propriétaire**: eAllman (Eric Allman)
- Occupation**: 12.17 Mo (81.11%) sur 15 Mo. A red 'Proche saturation' label is present. An 'Indicateur d'occupation' bar is shown.
- Tarification**: prix : 5.00€, Marge de dépassement : 2.00%, coût dépassement : 0.01€. A 'Tarif actuel du disque' label is present. A 'Modifier la tarification...' button is shown.
- Services**: Backup and LoadBalancer buttons. A 'Liste des services associés' label is present.

 At the bottom, there is a button 'Fermer et retourner à Mes disques' and a tooltip that says 'sur Click, appel de Disques/index'.

Élément	Indications
Contrôleur	Scan
Action	index
Paramètre	idDisque
Utilisateur connecté	L'utilisateur connecté est obtenu par l'appel de la méthode <b>Auth::getUser(\$controller)</b> où <b>\$controller</b> est le contrôleur actif.
Occupation, Quota	le quota est obtenu sur le tarif actuel du disque, la classe <b>ModelUtils</b> permet de connaître l'occupation en cours du disque
Composants visuels Bootstrap	on pourra utiliser les composants Phalcon-Jquery <b>htmlLabel</b> , <b>htmlGlyphButton</b> , <b>htmlListGroup</b>

L'indicateur d'occupation (**htmlLabel**) doit donner une indication sur le taux d'occupation en % :

Texte	Style	Valeurs
Peu occupé	info	de 0 à 10%
RAS	success	de 10 à 50%
Forte occupation	warning	de 50 à 80%
Proche saturation	danger	plus de 80%

### Conseils importants :

- Ajax est recommandé pour toutes les requêtes
- Penser à l'éventuel **{{script\_foot}}** dans les vues
- Préférer l'instanciation des composants phalcon JQuery en utilisant les méthodes de l'objet **\$this→jquery→bootstrap()**

## -- A poursuivre en dehors du TP...

### -- Création d'un disque (2 points)

#### //TODO 4.4.1

A partir de l'adresse **Disques/frm**

Afficher le formulaire de création d'un disque.

Soumettre le formulaire à l'adresse **Disques/update** pour créer le disque puis rediriger vers l'adresse **Scan/index/:idDisque**

Penser à le créer physiquement.

### -- Renommage disque (1 points)

A partir de l'adresse **Scan/index/:idDisque**

Caractéristiques du disque

**Nom : System**

Implémenter la modification du nom d'un Disque (avec renommage éventuel du dossier associé + vérification du nom)

### -- Contrôle d'accès (2 points)

URL	Accès	Résultat/message
<b>Disques/index</b>	Accessible uniquement pour un utilisateur connecté	Affiché message sur l'absence de connexion et proposer la connexion
<b>Scan/index/:idDisque</b>	Accessible uniquement pour un disque appartenant à l'utilisateur connecté	Afficher un message d'erreur "Accès à une ressource non autorisée"

Utiliser le [dispatcher phalcon](#) pour ce faire.

From:

<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:

<http://slamwiki2.kobject.net/slam4/php/phalcon/project/cloud>

Last update: **2019/08/31 14:21**

