

Dergal Nacer

Leroux Gwenael

Informatique et Réseaux

2<sup>ème</sup> année

# Manuel Utilisateur

## 1. Gestion des certificats

Pour pouvoir utiliser notre API cliente en HTTPS il faut tout d'abord importer le certificat "server.cer" auto-signé qui se trouve dans le répertoire "src" dans votre JDK ou JRE.

```
$JAVA_HOME/bin/keytool -import -v -trustcacerts -alias localhost  
-file server.cer -keystore $JAVA_HOME/lib/security/cacerts -keypass changeit  
-storepass changeit
```

## 2. L'intérêt de l'utilisation de Papaya DB

Papaya DB est un projet de système de gestion de base de données NOSQL développé par deux élèves, Nacer Dergal et Leroux Gwenaël, de l'ESIPE en formation Informatique et réseau.

L'objectif de Papaya DB est de permettre une recherche rapide dans une base de données. Pour se faire, Papaya DB dispose d'un cache contenant les base de données utilisées récemment. Les bases de données sont chargées en cache dès leur création ou dès la réalisation d'une requête dessus.

De plus lors de l'ajout d'un document dans une base de données, celui-ci serait indexé par rapport à tous ses champs. Les indexs sont rangés dans des arbres triés pour permettre une recherche optimale.

A la suppression d'un document, celui-ci sera enlevé de tous les indexs le concernant et il sera supprimé quand le nombre de document marqué à supprimer sera de plus de 80% du nombre de documents dans la base de données.

Ainsi lors d'une recherche, l'obtention des documents correspondant est théoriquement très rapide car tous les indexs auront déjà été mis à jour.

Dans le cas d'une recherche multi-critère, Papaya DB va séparer la recherche en critère et rechercher dans ses indexs les documents correspondant à chaque critère. Une fois cette opération finie, Papaya DB va regrouper les documents trouvés en excluant les doublons.

## 3. Les requêtes de Papaya DB

### a. La création d'une base de donnée (sécurisé par login:motdepasse) :

Pour créer une base de données dans Papaya DB, il faut exécuter une requête createDB. Si une base de données existe déjà avec ce nom, la requête fonctionnera comme même, même si aucune action ne sera réalisée au niveau du système de gestion de base de données.

### 1) Arguments :

Cette requête prend comme argument le nom de la base de donnée que vous souhaitez créer.

### 2) Valeur de retour :

Cette requête renvoie un String vide.

### 3) Exemples :

En utilisant notre programme de test :

CREATE DATABASE -> "nom de la base de donnée" login:motdepasse

Pour l'exemple le couple login:motdepasse est nacer:dergal

### b. La suppression d'une base de données (sécurisé par login:motdepasse):

Pour supprimer une base de données dans Papaya DB, il faut exécuter une requête dropDB.

### 1) Arguments :

Cette requête prend comme argument le nom de la base de donnée que vous souhaitez supprimer.

### 2) Valeur de retour :

Cette requête renvoie un String vide.

### 3) Erreurs possibles :

- La requête peut échouer si la base de données demandée n'existe pas
- La requête peut échouer si une erreur est apparue lors de la suppression de la base de données en mémoire

### 4) Exemples :

En utilisant notre programme de test :

DROP DATABASE -> "nom de la base de donnée" login:motdepasse

Pour l'exemple le couple login:motdepasse est nacer:dergal

### c. Obtenir tous les documents contenus d'une base de données (sécurisé):

Pour obtenir tous les documents contenus d'une base de données dans Papaya DB, il faut exécuter une requête getDB.

### 1) Arguments :

Cette requête prend comme argument le nom de la base de donnée dont vous souhaitez récupérer tous les documents.

### 2) Valeur de retour :

Cette requête renvoie un String contenant les JSON de tous les documents de la base de données.

### 3) Erreurs possibles :

- La requête peut échouer si la base de données demandée n'existe pas

### 4) Exemples :

En utilisant notre programme de test :

GET DATABASE -> "nom de la base de donnée"

## d. Ajouter un document dans une base de données :

Pour ajouter un document dans une base de données dans Papaya DB, il faut exécuter une requête create.

### 1) Arguments :

Cette requête prend comme argument le nom de la base de donnée où vous souhaitez ajouter le document suivi par le corps du document en String convertible en JSON.

Attention : le corps de votre document doit obligatoirement contenir le champ "name\_doc" correspondant au nom du document.

Attention : vous devez préciser le type de tous les champs. Pour cela, vous devrez donc rajouter la lettre correspondant au type souhaité au début du nom du champ. Par exemple: le champ 'name\_doc' qui contient un String deviendra 'sname\_doc'.

Liste des lettres de type actuellement :

- 's' = String
- 'i' = Integer
- 'd' = Date

### 2) Valeur de retour :

Cette requête renvoie un String vide.

### 3) Erreurs possibles :

- La requête peut échouer si la base de données demandée n'existe pas

- La requête peut échouer si une erreur est apparue lors de l'ajout du document dans la base de données en mémoire.
- La requête peut échouer si le corps du document ne contient pas le champ "name\_doc".
- La requête peut échouer si votre format Json n'est pas le bon.

#### 4) Exemples :

En utilisant notre programme de test :

CREATE -> "nom de la base de donnée" {"sname\_doc":"test","idate\_of\_creating":"50"}

#### e. Supprimer un document dans une base de données :

Pour supprimer un ou plusieurs document dans une base de données dans Papaya DB, il faut exécuter une requête drop.

##### 1) Arguments :

Cette requête prend comme argument le nom de la base de donnée où vous souhaitez supprimer le document suivi par le nom du document à supprimer.

Attention : Si plusieurs documents ont le nom que vous avez saisi alors ils seront tous supprimés.

##### 2) Valeur de retour :

Cette requête renvoie un String vide.

##### 3) Erreurs possibles :

- La requête peut échouer si la base de données demandée n'existe pas
- La requête peut échouer si une erreur est apparue lors de l'ajout du document dans la base de données en mémoire.

#### 4) Exemples :

En utilisant notre programme de test :

DROP -> "nom de la base de donnée" "nom du document"

## f. Obtenir l'ensemble des documents correspondant à une recherche dans une base de données :

Pour obtenir l'ensemble des documents correspondant à une recherche dans une base de données dans Papaya DB, il faut exécuter une requête get.

### 1) Arguments :

Cette requête prend comme argument le nom de la base de donnée où vous souhaitez rechercher les documents suivi par votre recherche.

Les opérations de comparaison possibles sont :

- '=' le champ visé dans le document doit être égal au critère
- '>' le champ visé dans le document doit être strictement supérieur au critère
- '>=' le champ visé dans le document doit être supérieur ou égal au critère
- '<' le champ visé dans le document doit être strictement inférieur au critère
- '<=' le champ visé dans le document doit être inférieur ou égal au critère
- 'between' le champ visé dans le document doit être compris entre les deux critères (Attention : le critère est compris dans l'intervalle)

Attention : pour une recherche multicritères, seul l'opérateur 'and' est implanté pour le moment. Celle-ci signifie l'union et non l'intersection entre l'ensemble des documents répondant au premier critère et ceux répondant au suivant. Vous devez séparer vos critères par 'and'.

### 2) Valeur de retour :

Cette requête renvoie un String contenant les JSON de tous les documents correspondants à votre recherche dans la base de données.

### 3) Erreurs possibles :

- La requête peut échouer si la base de données demandée n'existe pas
- La requête peut échouer si votre recherche n'est pas bien formée

### 4) Exemples :

En utilisant notre programme de test :

GET -> "nom de la base de donnée" "champ1"="test"and"champs2">"5"