

# Guide de l'utilisateur BPA Mise à niveau MongoDB v5.1

- [Introduction](#)
- [Mise à jour des versions](#)
- [Variables d'amortissement](#)
- [Suppression du rappel](#)
- [Mise à jour de la requête Suppression](#)
- [Mise à jour des modifications de requête](#)
- [Mise à jour de la fonction db.addUser\(\)](#)
- [Mise à jour des méthodes](#)
- [Mise à jour de GridFS](#)

## Introduction

BPA v4.1.2 inclut une mise à niveau vers MongoDB v7, car MongoDB v5 arrive en fin de vie. Ce document fournit des détails pour la mise à niveau de la version MongoDB dans les microservices.

## Mise à jour des versions

Mettez à jour les versions de package suivantes dans les microservices :

- «mongodb» : «^3.1.13 » à « mongodb » : « ^6.8.0 »
- «mangouste» : «5.8.7» à «mangouste» : « ^8.5.1 »

## Variables d'amortissement

Les variables suivantes sont désapprouvées dans MongoDB v7 :

- ssl
- useNewUrlParser
- utiliserTopologieUnifiée
- utiliserRechercherEtModifier
- utiliserCréerIndex
- sslValidate

Remplacez ssl par tls comme indiqué dans l'exemple suivant :

```
if (process.env.MONGO_SSL == "true") {  
  //options.ssl = true;  
  options.tls = true;  
  options.tlsAllowInvalidCertificates = true;  
}
```

## Suppression du rappel

Le rappel dans les requêtes MongoDB a été déconseillé dans MongoDB v7. La requête mise à jour s'affiche en vert ci-dessous.

```
deviceCredentials.find(query, ['name', 'credential_id', 'credential_type', 'description'], function (error, credentials) {  
  deviceCredentials.find(query, ['name', 'credential_id', 'credential_type', 'description']).then(credentials => {  
    let result = [];
```

Requête mise à jour

## Mise à jour de la requête Suppression

Après avoir exécuté la requête de suppression, la variable de réponse doit utiliser deletedCount à la place de n (par exemple, remplacer result.n par result.deletedCount).

```
if (!result.n) {  
  if (!result.deletedCount) {  
    return res.status(404).json(errorHandlerObject.errorHandler(false, false, errorString, 404)).end('');  
  }  
}
```

Remplacer le résultat.n

## Mise à jour des modifications de requête

Après avoir effectué l'opération de mise à jour, la variable de réponse doit utiliser modifiedCount à la place de n/nModified (par exemple, Replace devices.n / devices.nModified to devices.modifiedCount dans comme indiqué ci-dessous).

```
if (devices && devices.n && devices.n > 0) {  
  if (devices && devices.modifiedCount && devices.modifiedCount > 0) {  
    response.success.push({ name: element.name, message: UPDATE_ASSETS.SUCCESS });  
  }  
}
```

Remplacer périphérique.n ou périphériques.nModifié

## Mise à jour de la fonction db.addUser()

Remplacez la fonction db.addUser() par db.command(createUser: «username») comme indiqué dans l'exemple suivant.

```
migration_db.addUser(process.env.MONGODB_USER, process.env.MONGODB_PASSWORD, {
  roles: [{
    role: 'readWrite',
    db: database_name
  }]
})
migration_db.command({ // in mongodb 6.x addUser function removed
  createUser: process.env.MONGODB_USER,
  pwd: process.env.MONGODB_PASSWORD,
  roles: [ { role: 'readWrite', db: database_name } ]
}). then( (result) => {
```

Remplacer db.addUser()

## Mise à jour des méthodes

Mettez à jour les méthodes suivantes déconseillées dans MongoDB v7 comme indiqué ci-dessous.

Méthode déconseillée	Nouvelle méthode
mise à jour	updateOne ou updateMany
deposer	delete ou deleteMany
chef d'accusation	countDocuments
rechercherUnEtSupprimer	RechercherUnEtSupprimer

## Mise à jour de GridFS

Dans MongoDB v7, la classe GridFSBucket est utilisée pour interagir avec GridFS, une spécification de stockage et de récupération de fichiers volumineux. La classe GridFSBucket fournit des méthodes pour télécharger, télécharger et gérer des fichiers dans GridFS.

Remplacez gridfs-stream en important GridFSBucket de MongoDB. Reportez-vous aux images

suivantes pour afficher les modifications.

```
const Gridfs = require('gridfs-stream');  
const { GridFSBucket } = require('mongodb');
```

Remplacer gridfs-stream

```
let dbConn = mongoose.connection.db;  
let dbDriver = mongoose.mongo;  
let gridFs = new Gridfs(dbConn, dbDriver);  
let readStream = gridFs.createReadStream({ _id: pdfrefId });  
let gridFs = new GridFSBucket(dbConn);  
let fileId = new mongoose.Types.ObjectId(pdfrefId);  
let readStream = gridFs.openDownloadStream(fileId);
```

Exemple 1

```

let dbConn = mongoose.connection.db;
let dbDriver = mongoose.mongo;
let gridFs = new Gridfs(dbConn, dbDriver);
let gridFs = new GridFSBucket(dbConn);

let writeStream = gridFs.createWriteStream({
  filename: fileName,
  mode: 'w',
  content_type: contentType
let writeStream = gridFs.openUploadStream(fileName, {
  contentType: contentType
});
fs.createReadStream(fileName).pipe(writeStream);
writeStream.on('close', async (file) => {

writeStream.on('finish', async (file) => {

if (storeType === 'pdf') {
  reportObj.pdfrefId = file._id;
  reportObj.pdfrefId = writeStream.id;
  reportObj.pdfGenerationState = PDF_GENERATE_STATE.COMPLETED;
} else reportObj.htmlrefId = file._id;
} else reportObj.htmlrefId = writeStream.id;

});

```

Exemple 2

À propos de cette traduction

Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.