BPA-Benutzerhandbuch MongoDB Upgrade v5.1

Einleitung

BPA v4.1.2 beinhaltet ein Upgrade auf MongoDB v7, da sich MongoDB v5 dem Ende des Lebenszyklus nähert. Dieses Dokument enthält Details zum Upgrade der MongoDB-Version in Microservices.

Aktualisieren von Versionen

Aktualisieren Sie die folgenden Paketversionen in Microservices:

- ‚mongodb': "^3.1.13" zu "mongodb": "^6.8.0"
- "Mungo": "5.8.7" zu "Mungo": "^8.5.1"

Abschreibungsvariablen

Die folgenden Variablen sind in MongoDB v7 veraltet:

- SSL
- useNewUrlParser
- VerwendenEinheitlicheTopologie
- VerwendenSuchenUndÄndern
- ErstellenIndex
- sslValidieren

Ersetzen Sie ssl durch tls, wie im folgenden Beispiel gezeigt:

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

Rückruf wird entfernt

Der Rückruf in MongoDB-Abfragen wurde in MongoDB v7 veraltet. Die aktualisierte Abfrage wird unten grün angezeigt.

```
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 = [];
```

Aktualisierte Abfrage

Aktualisieren der Löschabfrage

Nach Durchführung der Löschabfrage sollte die Antwortvariable deletedCount anstelle von n verwenden (z. B. result.n durch result.deletedCount ersetzen).

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

Ergebnis ersetzen.n

Aktualisieren von Abfrageänderungen

Nach der Durchführung des Aktualisierungsvorgangs sollte die Antwortvariable modifiedCount anstelle von n/nModified verwenden (z. B. Replace devices.n / devices.nModified to devices.modifiedCount in wie unten gezeigt).

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

Ersetzen Sie device.n oder devices.nModified

db.addUser()-Funktion wird aktualisiert

Ersetzen Sie die db.addUser()-Funktion durch db.command(createUse: "username"), wie im folgenden Beispiel gezeigt.

```
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) => {
```

db.addUser() ersetzen

Methoden aktualisieren

Aktualisieren Sie die folgenden Methoden veraltet in MongoDB v7 wie unten angegeben.

Veraltete Methode	Neue Methode
update	updateOne oder updateMany
entfernen	Löschen oder LöschenViele
Zählung	AnzahlDokumente
FindOneAndRemove	FindOneUndLöschen

Aktualisieren von GridFS

In MongoDB v7 wird die GridFSBucket-Klasse verwendet, um mit GridFS zu interagieren, einer Spezifikation zum Speichern und Abrufen großer Dateien. Die GridFSBucket-Klasse stellt Methoden zum Hochladen, Herunterladen und Verwalten von Dateien in GridFS bereit.

Ersetzen Sie gridfs-stream, indem Sie GridFSBucket von MongoDB importieren. Die Änderungen können Sie den folgenden Bildern entnehmen.

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

Gridfs-Stream ersetzen

```
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);
```

Beispiel 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;

});
```

Beispiel 2

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.