

BPA-gebruikershandleiding MongoDB Upgrade v5.1

- [Inleiding](#)
- [Versies bijwerken](#)
- [afschrijvingsvariabelen](#)
- [Terugbellen verwijderen](#)
- [De verwijderquery bijwerken](#)
- [Wijzigingen in zoekopdracht bijwerken](#)
- [De functie db.addUser\(\) bijwerken](#)
- [Methoden bijwerken](#)
- [GridFS bijwerken](#)

Inleiding

BPA v4.1.2 bevat een upgrade naar MongoDB v7, omdat MongoDB v5 het einde van zijn levensduur nadert. Dit document bevat details voor het upgraden van de MongoDB-versie in microservices.

Versies bijwerken

Werk de volgende pakketversies bij in microservices:

- "Mongodb": "^3.1.13" naar "Mongodb": "^6.8.0"
- "Mongoose": "5.8.7" naar "Mongoose": "^8.5.1"

afschrijvingsvariabelen

De volgende variabelen worden afgeschreven in MongoDB v7:

- SSL
- NewUrlParser gebruiken
- gebruikUnifiedTopology
- Zoeken en wijzigen gebruiken
- useCreateIndex
- SSL valideren

Vervang SSL door tls zoals in het volgende voorbeeld:

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

Terugbellen verwijderen

Callback in MongoDB-query's is afgeschreven in MongoDB v7. De bijgewerkte query wordt hieronder in het groen weergegeven.

```
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 = [];
```

Bijgewerkte zoekopdracht

De verwijderquery bijwerken

Na het uitvoeren van de verwijderquery moet de responsvariabele deletedCount gebruiken in plaats van n (bijv. result.n vervangen door result.deletedCount).

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

Resultaat vervangen.n

Wijzigingen in zoekopdracht bijwerken

Nadat de update is uitgevoerd, moet de responsvariabele modifiedCount gebruiken in plaats van n/nModified (bijv. Replace devices.n / devices.nModified to devices.modifiedCount in zoals hieronder wordt weergegeven).

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

Device.n of devices.nModified vervangen

De functie db.addUser() bijwerken

Vervang de functie db.addUser() door db.command(createUser: "username") zoals in het volgende voorbeeld wordt weergegeven.

```
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) => {
```

Vervang db.addUser()

Methoden bijwerken

Werk de volgende methoden bij die zijn afgekeurd in MongoDB v7, zoals hieronder wordt aangegeven.

afgekeurde methode	Nieuwe methode
bijwerken	updateOne of updateMany
wegnemen	verwijderen of verwijderenVeel
tellen	countDocuments
FindOneAndRemove	findOneAndDelete

GridFS bijwerken

In MongoDB v7 wordt de klasse GridFSBucket gebruikt om te communiceren met GridFS, een specificatie voor het opslaan en ophalen van grote bestanden. De klasse GridFSBucket biedt methoden voor het uploaden, downloaden en beheren van bestanden in GridFS.

Vervang grids-stream door GridFSBucket te importeren van MongoDB. Raadpleeg de volgende

afbeeldingen om de wijzigingen te bekijken.

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

Rasters-stroom vervangen

```
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);
```

Voorbeeld 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;

});

```

Voorbeeld 2

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.