Guida per l'utente BPA Aggiornamento MongoDB versione 5.1

Introduzione

BPA v4.1.2 include un aggiornamento a MongoDB v7, in quanto MongoDB v5 sta per giungere alla fine del ciclo di vita. Questo documento fornisce i dettagli per l'aggiornamento della versione di MongoDB in microservizi.

Aggiornamento delle versioni

Aggiornare le seguenti versioni dei pacchetti in microservizi:

- "mongodb": "^3.1.13" a "mongodb": "^6.8.0"
- "mangusta": "5.8.7" per "mangusta": "^8.5.1"

Variabili di ammortamento

Le seguenti variabili sono deprecate in MongoDB v7:

- SSL
- usaNuovoParserUrl
- usaTopologiaUnificata
- usaTrovaModifica
- usaCreaIndice
- ConvalidaSSL

Sostituire ssl con tls, come mostrato nell'esempio seguente:

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

Rimozione della richiamata

Il callback nelle query MongoDB è deprecato in MongoDB v7. La query aggiornata viene visualizzata in verde.

```
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 = [];
```

Query aggiornata

Aggiornamento della query di eliminazione

Dopo aver eseguito la query di eliminazione, la variabile di risposta deve utilizzare deletCount al posto di n (ad esempio, sostituire result.n con result.DeletCount).

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

Sostituisci risultato.n

Aggiornamento delle modifiche alle query

Dopo aver eseguito l'operazione di aggiornamento, la variabile di risposta deve utilizzare modifiedCount al posto di n/nModified (ad esempio, Replace devices.n / devices.nModified to devices.modifiedCount come mostrato di seguito).

```
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 {
```

Sostituire device.n o devices.nModified

Aggiornamento della funzione db.addUser()

Sostituire la funzione db.addUser() con db.command(createUse: "username") come mostrato nell'esempio che segue.

```
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) => {
```

Sostituisci db.addUser()

Aggiornamento dei metodi

Aggiornare i seguenti metodi deprecati in MongoDB v7 come indicato di seguito.

Metodo deprecato	New, metodo
update	updateOne o updateMany
rimuovere	delete o deleteMany
conteggio	conteggioDocumenti
trovaUnaERimuovi	TrovaEdElimina

Aggiornamento di GridFS

In MongoDB v7, la classe GridFSBucket viene utilizzata per interagire con GridFS, una specifica per la memorizzazione e il recupero di file di grandi dimensioni. La classe GridFSBucket fornisce metodi per caricare, scaricare e gestire file in GridFS.

Sostituire gridfs-stream importando GridFSBucket da MongoDB. Fare riferimento alle immagini seguenti per visualizzare le modifiche.

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

Sostituisci flusso griglia

```
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);
```

Esempio 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;

});
```

Esempio 2

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).