BPAユーザガイドMongoDBアップグレードv5.1

- はじめに
- バージョンの更新
- 減価償却変数
- コールバックの削除
- 削除クエリの更新
- クエリーの変更の更新
- db.addUser()関数の更新
- メソッドの更新
- GridFSの更新

はじめに

BPA v4.1.2には、MongoDB v5のサポート終了が近づいているため、MongoDB v7へのアップグレードが含まれています。このドキュメントでは、マイクロサービスのMongoDBバージョンのアップグレードの詳細について説明します。

バージョンの更新

マイクロサービスで次のパッケージバージョンを更新します。

- "mongodb": "^3.1.13" ~ "mongodb": "^6.8.0"
- 「マングース」:「5.8.7」から「マングース」:「^8.5.1」

減価償却変数

次の変数は、MongoDB v7では廃止されています。

- SSL
- useNewUrlParser
- 統合トポロジの使用
- 検索と変更の使用
- 作成インデックスの使用
- ssl検証

次の例に示すように、sslをtlsに置き換えます。

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

コールバックの削除

MongoDB v7では、MongoDBクエリのコールバックは廃止されました。更新されたクエリが下に緑色で表示されます。

```
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 = [];
```

更新されたクエリ

削除クエリの更新

削除クエリの実行後、応答変数はnの代わりにdeletedCountを使用する必要があります(例：result.nをresult.deletedCountで置き換えます)。

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

結果を置き換えます。n

クエリーの変更の更新

更新操作の実行後、応答変数はn/nModifiedの代わりにmodifiedCountを使用します(例：以下に示すように、devices.n / devices.nModifiedをdevices.modifiedCountに置き換えます)。

```
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 {
```

device.nまたはdevices.nModifiedを置き換えます

db.addUser()関数の更新

次の例に示すように、db.addUser()関数をdb.command(createUse: "username")に置き換えます。

```
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()の置換

メソッドの更新

次に示すように、MongoDB v7で廃止された次のメソッドを更新します。

非推奨のメソッド	新しいメソッド
アップデート	updateOneまたはupdateMany
削除	deleteまたはdeleteMany
count	countDocuments
findOneAndRemove	findOneAndDelete

GridFSの更新

MongoDB v7では、GridFSBucketクラスを使用して、大きなファイルを格納および取得するための仕様であるGridFSと対話します。GridFSBucketクラスは、GridFSでファイルをアップロード、ダウンロード、および管理するためのメソッドを提供します。

MongoDBからGridFSBucketをインポートして、gridfs-streamを置き換えます。変更を確認するには、次の図を参照してください。

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

gridfs-streamの置き換え

```
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);
```

例 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;

});
```

例 2

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。