

BPA 사용 설명서 MongoDB 업그레이드 v5.1

- [소개](#)
- [버전 업데이트](#)
- [감가상각 변수](#)
- [콜백 제거](#)
- [삭제 쿼리 업데이트](#)
- [쿼리 변경 사항 업데이트](#)
- [db.addUser\(\) 함수 업데이트](#)
- [방법 업데이트](#)
- [GridFS 업데이트](#)

소개

BPA v4.1.2에는 MongoDB v5가 단종될 예정이므로 MongoDB v7로의 업그레이드가 포함됩니다. 이 문서에서는 microservices에서 MongoDB 버전을 업그레이드하는 방법을 자세히 설명합니다.

버전 업데이트

microservices에서 다음 패키지 버전을 업데이트합니다.

- "mongodb": "^3.1.13"에서 "mongodb"로: "^6.8.0"
- "mongoose": "5.8.7"에서 "mongoose"로: "^8.5.1"

감가상각 변수

다음 변수는 MongoDB v7에서 더 이상 사용되지 않습니다.

- ssl
- useNewUri파서
- 유니파이드토폴로지 사용
- 찾기 및 수정 사용
- 인덱스만들기사용
- sslValidate

다음 예에 표시된 대로 ssl을 tls로 교체합니다.

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

콜백 제거

MongoDB 쿼리의 콜백은 MongoDB v7에서 더 이상 사용되지 않습니다. 업데이트된 쿼리는 아래 녹색으로 표시됩니다.

```
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 = [];
```

업데이트된 쿼리

삭제 쿼리 업데이트

삭제 쿼리를 수행한 후 응답 변수는 n 대신 deletedCount를 사용해야 합니다(예: result.n을 result.deletedCount로 대체).

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

결과 바꾸기.n

쿼리 변경 사항 업데이트

업데이트 작업을 수행한 후 응답 변수는 n/nModified 대신 modifiedCount를 사용해야 합니다(예: 아래와 같이 Replace devices.n / devices.nModified to devices.modifiedCount in).

```
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 또는 devices.nModified 교체

db.addUser() 함수 업데이트

db.addUser() 함수를 db.command(createUser: 다음 예에 표시된 대로 "username")을 입력합니다.

```
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() 바꾸기

방법 업데이트

아래에 표시된 대로 MongoDB v7에서 더 이상 사용되지 않는 다음 메서드를 업데이트합니다.

사용되지 않는 메서드	새 방법
update	updateOne 또는 updateMany
제거	삭제 또는 삭제모두
카운트	count문서
찾기모두 제거	찾기 및 삭제

GridFS 업데이트

MongoDB v7에서는 GridFSBucket 클래스를 사용하여 대용량 파일을 저장하고 검색하는 사양인 GridFS와 상호 작용합니다. GridFSBucket 클래스는 GridFS에서 파일을 업로드, 다운로드 및 관리하는 메서드를 제공합니다.

MongoDB에서 GridFSBucket을 가져와서 gridfs-stream을 대체합니다. 변경 사항을 보려면 다음 이미지를 참조하십시오.

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

Gridfs 스트림 교체

```
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);
```

예 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;

});

```

예 2

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.