

Guia do usuário do BPA Atualização do MongoDB v5.1

- [Introdução](#)
- [Atualizando versões](#)
- [Variáveis de Depreciação](#)
- [Removendo Retorno de Chamada](#)
- [Atualizando a Consulta de Exclusão](#)
- [Atualizando Alterações de Consulta](#)
- [Atualizando a Função db.addUser\(\)](#)
- [Atualizando métodos](#)
- [Atualizando GridFS](#)

Introdução

O BPA v4.1.2 inclui uma atualização para o MongoDB v7, pois o MongoDB v5 está se aproximando do fim da vida útil. Este documento fornece detalhes para atualizar a versão MongoDB em microsserviços.

Atualizando versões

Atualize as seguintes versões de pacote em microsserviços:

- "mongodb": "^3.1.13" em "mongodb": "^6.8.0"
- "Mangusto": "5.8.7" a "manganso": "^8.5.1"

Variáveis de Depreciação

As seguintes variáveis foram preteridas no MongoDB v7:

- ssl
- usarNovoAnalisadorUrl
- usarTopologiaUnificada
- usarLocalizarEModificar
- usarCriarÍndice
- sslValidate

Substitua ssl por tls conforme mostrado no exemplo a seguir:

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

Removendo Retorno de Chamada

O retorno de chamada em consultas MongoDB foi preterido no MongoDB v7. A consulta atualizada é exibida em verde abaixo.

```
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 = [];
```

Consulta Atualizada

Atualizando a Consulta de Exclusão

Após executar a consulta delete, a variável de resposta deve usar deletedCount no lugar de n (por exemplo, substituir result.n por result.deletedCount).

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

Substituir result.n

Atualizando Alterações de Consulta

Após executar a operação de atualização, a variável de resposta deve usar modifiedCount no lugar de n/nModified (por exemplo, Substituir devices.n / devices.nModified a devices.modifiedCount, como mostrado abaixo).

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

Substitua device.n ou devices.nModified

Atualizando a Função db.addUser()

Substitua a função db.addUser() por db.command(createUser: "username") conforme mostrado no exemplo a seguir.

```
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) => {
```

Substitua db.addUser()

Atualizando métodos

Atualize os seguintes métodos preteridos no MongoDB v7 conforme indicado abaixo.

Método preterido	Novo método
update	updateOne ou updateMany
remove	delete ou deleteMany
contagem	countDocuments
localizarUmERemover	localizarUmaExcluir

Atualizando GridFS

No MongoDB v7, a classe GridFSBucket é usada para interagir com o GridFS, uma especificação para armazenar e recuperar arquivos grandes. A classe GridFSBucket fornece métodos para carregar, baixar e gerenciar arquivos no GridFS.

Substitua gridfs-stream importando GridFSBucket do MongoDB. Consulte as imagens a seguir para visualizar as alterações.

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

Substituir gridfs-stream

```
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);
```

Exemplo 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;

});

```

Exemplo 2

Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.