

# GridFS in Spring Data MongoDB

## 1. Overview

This tutorial will explore one of the **core features of Spring Data MongoDB: interacting with *GridFS***.

The GridFS storage spec is mainly used for working with files that exceed the *BSON*-document size limit of 16MB. And Spring Data provides a *GridFsOperations* interface and its implementation – *GridFsTemplate* – to easily interact with this filesystem.

## 2. Configuration

### 2.1. XML Configuration

Let's start with the simple XML configuration for the *GridFsTemplate*:

```
1<bean id="gridFsTemplate"
  class="org.springframework.data.mongodb.gridfs.GridFsTemplate">
2    <constructor-arg ref="mongoDbFactory" />
3    <constructor-arg ref="mongoConverter" />
4</bean>
```

The constructor arguments to the *GridFsTemplate* include bean references to *mongoDbFactory*, which creates a Mongo database, and *mongoConverter*, which converts between Java and MongoDB types. Their bean definitions are below.

```
1<mongo:db-factory id="mongoDbFactory" dbname="test" mongo-ref="mongo" />
2
3<mongo:mapping-converter id="mongoConverter" base-
  package="org.baeldung.converter">
4    <mongo:custom-converters base-package="org.baeldung.converter"/>
5</mongo:mapping-converter>
```

### 2.2. Java Configuration

Let's create a similar configuration, only with Java:

```
1@Bean
2public GridFsTemplate gridFsTemplate() throws Exception {
3    return new GridFsTemplate(mongoDbFactory(), mappingMongoConverter());
4}
```

For this configuration we used *mongoDbFactory* and *mappingMongoConverter* from *org.springframework.data.mongodb.config.AbstractMongoConfiguration*.

### 3. *GridFsTemplate* Core Methods

#### 3.1. *store*

The *store* method saves a file into MongoDB.

Suppose we have an empty database and wish to store a file in it:

```
InputStream inputStream = new
1FileInputStream("src/main/resources/test.png");

2String id =

3 gridFsTemplate.store(inputStream, "test.png", "image/png",
  metaData).getId().toString();
```

Note that we can save additional metadata along with the file by passing a *DBObject* to the *store* method. For our example, the *DBObject* might look something like this:

```
1DBObject metaData = new BasicDBObject();

2metaData.put("user", "alex");
```

*GridFS* uses two collections to store the file metadata and its content. The file's metadata is stored in the *files* collection, and the file's content is stored in the *chunks* collection. Both collections are prefixed with *fs*.