2.7. GridFS

2.7.1. Writing

GridFS support comes in the form of the `FileField` field object. This field acts as a file-like object and provides a couple of different ways of inserting and retrieving data. Arbitrary metadata such as content type can also be stored alongside the files. The object returned when accessing a FileField is a proxy to Pymongo's GridFS In the following example, a document is created to store details about animals, including a photo:

```
class Animal(Document):
    genus = StringField()
    family = StringField()
    photo = FileField()

marmot = Animal(genus='Marmota', family='Sciuridae')

with open('marmot.jpg', 'rb') as fd:
    marmot.photo.put(fd, content_type = 'image/jpeg')
marmot.save()
```

2.7.2. Retrieval

So using the `FileField` is just like using any other field. The file can also be retrieved just as easily:

```
marmot = Animal.objects(genus='Marmota').first()
photo = marmot.photo.read()
content_type = marmot.photo.content_type
```

Note

If you need to read() the content of a file multiple times, you'll need to "rewind" the file-like object using *seek*:

```
marmot = Animal.objects(genus='Marmota').first()
content1 = marmot.photo.read()
assert content1 != ""

content2 = marmot.photo.read()    # will be empty
assert content2 == ""

marmot.photo.seek(0)              # rewind the file by setting the current position
of the cursor in the file to 0
content3 = marmot.photo.read()
assert content3 == content1
```

2.7.3. Streaming

Streaming data into a `FileField` is achieved in a slightly different manner. First, a new file must be created by calling the `new_file()` method. Data can then be written using `write()`:

```
marmot.photo.new_file()
marmot.photo.write('some_image_data')
marmot.photo.write('some_more_image_data')
marmot.photo.close()

marmot.save()
```

2.7.4. Deletion

Deleting stored files is achieved with the `delete()` method:

```
marmot.photo.delete()    # Deletes the GridFS document
marmot.save()            # Saves the GridFS reference (being None) contained in the
marmot instance
```

Warning

The FileField in a Document actually only stores the ID of a file in a separate GridFS collection. This means that deleting a document with a defined FileField does not actually delete the file. You must be careful to delete any files in a Document as above before deleting the Document itself.

2.7.5. Replacing files

Files can be replaced with the `replace()` method. This works just like the `put()` method so even metadata can (and should) be replaced:

```
another_marmot = open('another_marmot.png', 'rb')
marmot.photo.replace(another_marmot, content_type='image/png')  # Replaces the GridFS
document
marmot.save()                                                   # Replaces the GridFS
reference contained in marmot instance
```