



Technology Series

MEAN

Full Stack

02

Presented by
Subhash EP



Subhash

me A Full Stack



Subhash

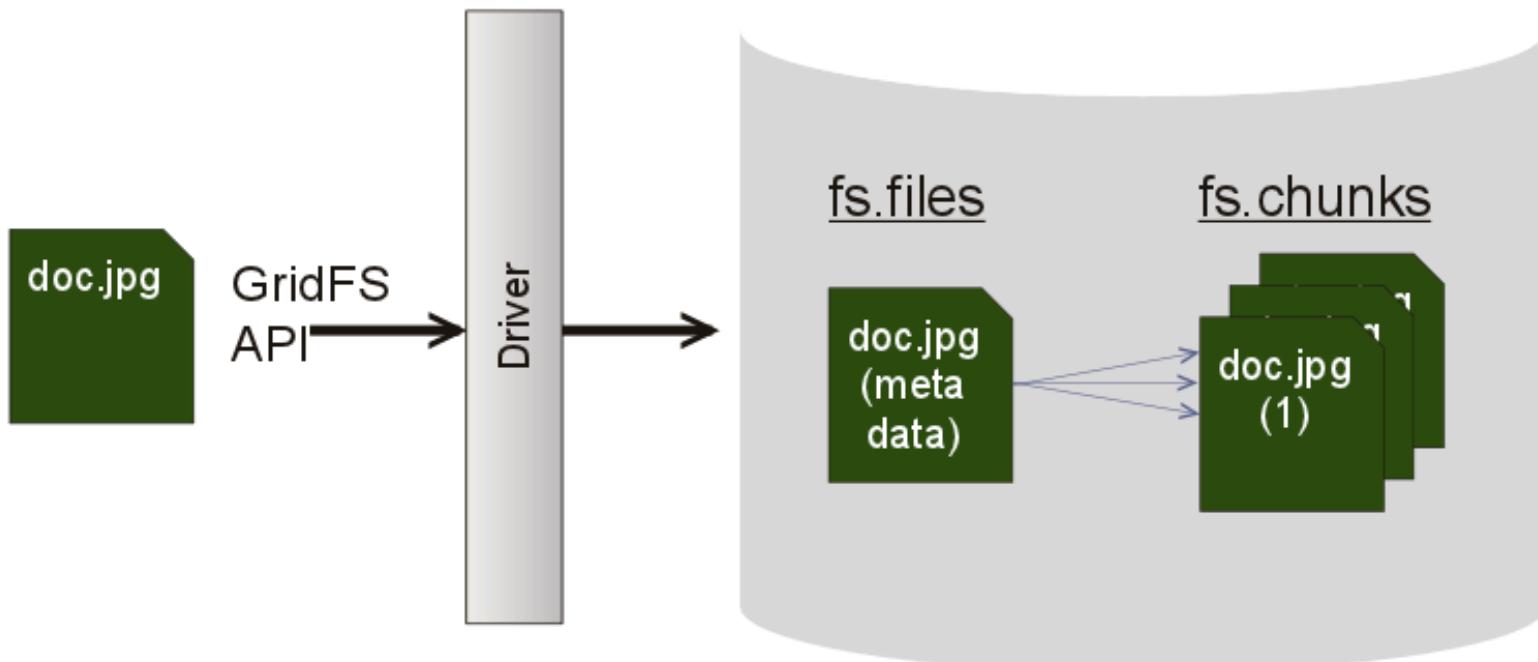
Schema

- a schema is a way to describe the structure of data

Schema Data Types

String	Buffer
Number	ObjectId
Date	Mixed
Boolean	Array

GridFS API



Sample Schema

```
var userSchema = new mongoose.Schema({  
    name: String,  
    email: {type: String, unique:true},  
    createdOn: { type: Date, default: Date.now },  
    modifiedOn: Date,  
    lastLogin: Date  
});
```

Typical Document in DB

- (line breaks are added for readability):

```
{ "____v" : 0,  
  "_id" : ObjectId("5126b7a1f8a44d1e32000001"),  
  "createdOn" : ISODate("2014-02-22T00:11:13.436Z"),  
  "email" : "subhash@vsnl.in",  
  "lastLogin" : ISODate("2014-04-03T12:54:42.734Z"),  
  "modifiedOn" :  
    ISODate("2014-04-03T12:56:26.009Z"),  
  "name" : "Subhash Edakkuda" }
```

v

- internal versioning number automatically set by Mongoose
- It doesn't increment when a document is changed,
 - but instead is automatically incremented whenever an array within the document is updated
 - in such a way that might cause the indexed position of some of the entries to have changed.

Building Models

- a model is a compiled version of the schema.
- A single instance of a model maps directly to a single document in the database.
- With this 1:1 relationship,
 - it is the model that handles all document interaction
 - creating, reading, saving, and deleting.

Sample Model in our Project

- `mongoose.model('User', userSchema);`
- The name of the model
- The name of the schema to compile

Instances

- have a good understanding of how a model works.
- After building the User model, create two instances.

```
var userOne = new User({ name: 'Subhash' });  
var userTwo = new User({ name: 'Divya' });
```

Save

```
userTwo.save(function (err) {  
  if (err) return handleError(err);  
  // userTwo (Divya) is saved!  
});  
  
userOne.save(function (err) {  
  if (err) return handleError(err);  
  // userOne (Subhash) is saved!  
});
```

FindOne

```
User.findOne({'name' : 'Divya',
function(err,user) {
if(!err){
  console.log(user);
}
});
```

Find many

```
User.find({}, function(err, users) {  
  if(!err){  
    console.log(users);  
  }  
});
```

Overriding collection name in schema

```
var userSchema = new mongoose.Schema({  
    name: String,  
    email: {  
        type: String, unique:true  
    }  
},  
{  
    collection: 'myuserlist'  
});
```

Overriding collection name in model

```
mongoose.model( 'User', userSchema,  
'myuserlist' );
```

That's all



End of Session

©

subhash.ep@gmail.com
linkedin.com/in/subhashep