Models

Models are fancy constructors compiled from our `Schema` definitions. Instances of these models represent documents which can be saved and retrieved from our database. All document creation and retrieval from the database is handled by these models.

Compiling your first model

```
var schema = new mongoose.Schema({ name: 'string',
var Tank = mongoose.model('Tank', schema);
```

The first argument is the *singular* name of the collection your model is for. **Mongoose automatically looks for the *plural* version of your model name.** Thus, for the example above, the model Tank is for the **tanks** collection in the database. The `.model()` function makes a copy of `schema`. Make sure that you've added everything you want to `schema` before calling `.model()`!

Constructing documents

Documents are instances of our model. Creating them and saving to the database is easy:

```
var Tank = mongoose.model('Tank', yourSchema);

var small = new Tank({ size: 'small' });
small.save(function (err) {
  if (err) return handleError(err);
  // saved!
})

// or
```

```
Tank.create({ size: 'small' }, function (err, sma
  if (err) return handleError(err);
  // saved!
})
```

Note that no tanks will be created/removed until the connection your model uses is open. Every model has an associated connection. When you use `mongoose.model()`, your model will use the default mongoose connection.

```
mongoose.connect('localhost', 'gettingstarted');
```

If you create a custom connection, use that connection's `model()` function instead.

```
var connection = mongoose.createConnection('mongo
var Tank = connection.model('Tank', yourSchema);
```

Querying

Finding documents is easy with Mongoose, which supports the rich query syntax of MongoDB. Documents can be retreived using each `models` find, findById, findOne, or where static methods.

```
Tank.find({ size: 'small' }).where('createdDate')
```

See the chapter on querying for more details on how to use the Query api.

Removing

mongoose

- home
- FAQ
- plugins
- change log
- support
- fork
- guide
  - schemas
    - types
    - custom
  - models

Models have a static `remove` method available for removing all documents matching `conditions`.

```
Tank.remove({ size: 'large' }, function (err) {
  if (err) return handleError(err);
  // removed!
});
```

Updating

Each `model` has its own `update` method for modifying documents in the database without returning them to your application. See the API docs for more detail.

If you want to update a single document in the db and return it to your application, use findOneAndUpdate instead.

Yet more

The API docs cover many additional methods available like count, mapReduce, aggregate, and more.

Next Up

Now that we've covered `Models`, let's take a look at Documents.