

Schemas & Models

NEXT: [ROUTES AND CONTROLLERS](#) ➔

To interact with a MongoDB database via mongoose expressively, you need to define schemas and models. They just tell mongoose what the data structure looks like and the data types. This way, you have a more structured and organized way of writing to the database.

To define a schema, we make use of the `mongoose.Schema` method:

```
JS
// ./model/index.js
const mongoose = require('mongoose');
const Schema = mongoose.Schema,
    model = mongoose.model.bind(mongoose),
    ObjectId = mongoose.Schema.Types.ObjectId;

const productSchema = Schema({
  id: ObjectId,
  name: String,
  image: String,
  price: Number,
  description: String,
  // One to many relationship
  manufacturer: {type: ObjectId, ref: 'Manufacturer'}
});

const manufacturerSchema = Schema({
  id: ObjectId,
  name: String,
});
```

A schema takes an object that describes the data structure and types. There is a special type known as `ObjectId` which serves as unique ID for our stored items.

You can also create a one to many relationship using the `ref` property as shown in the example above. The `ref` value must be the model that we are setting the relationship with.

Speaking of models, let's create some for the defined schemas:

```
JS
const Product = model('Product', productSchema);
const Manufacturer = model('Manufacturer', manufacturerSchema)
```

You can export the model to be used by other parts of our project:

```
JS
module.exports = {Product, Manufacturer};
```

Getting Started		
1	Introduction	Free
2	Tools	Free
3	Our Task	Free
4	Hello World	Free
Vue Basics		
5	Vue CLI	Free
6	Components and the Vue Instance	Free
7	Template Syntax	Free
8	Conditional & List Rendering	Free
9	Styles & Classes	Free
Routing		
10	Single Page Apps	Free
11	Creating Routes	Free
12	Route Outlets & Links	Free
13	Nested & Dynamic Routes	Free
Forms		
14	Two Way Binding (Reactivity)	Free
15	Form Validation	Free
16	Handling Form Submissions	Free
API Backend		
17	Prelude	Free
18	Setup	Free
19	Provisioning a MongoDB Database	Free
20	Enabling CORS	Free
21	Schemas & Models	Free
22	Routes and Controllers	Free
23	Testing with POSTman	Free
Vuex		
24	The Current Problem	Free
25	State Management	Free
26	Getters	Free
27	Mutations	Free
28	Actions	Free
Using Store In Components		
29	State and Actions on Components	Free
30	LAB: Product List	Free
31	LAB: Product Details	Free
32	LAB: Admin Features	Free
33	LAB: Spinner, Cart & Store Subscription	Free