

A photograph of a rustic wooden porch or walkway. On the right, there is a wooden building with horizontal siding. The porch is made of wooden planks and has several vertical wooden posts. In the background, there are palm trees and a desert landscape under a clear sky. The text "Mongoose and ORMs (or ODMs)" is overlaid in white serif font in the center of the image.

Mongoose and ORMs (or ODMs)

What is an ORM/ODM?

- ORM – Object Relational Mapper
 - ODM – Object Document Mapper
- O{R,D}Ms are a way of mapping data to native objects
- Typically provide some way to match schemas to data types
- Basic validation
- Abstraction over relations
- Query abstractions

Why an O{R,D}M over lower level connectors/clients?

- Richer APIs
 - Rather than requiring long strings of raw objects or text, we get simplified methods
 - More typesafe(ish) functions
- Model/document validation
 - We can specify our schema upfront and validate in software
 - Avoids pushing validation to the database and allows for app-level validation
- Better relation modeling
 - Instead of complex queries, simple methods that abstract over them
 - More explicit way to write up object/document references

What is Mongoose?

- MongoDB ODM
- Has a straightforward API mostly centered around the `Model` object
- `Model`'s have `Schema`'s that allow for specifying data types
 - Sort of a schema without enforcing it at the database level
- Query interface that makes interacting with data simpler

How do I use it?

```
1  npm install mongoose
```

```
1  import mongoose from 'mongoose';  
2  
3  await mongoose.connect('mongodb://localhost:27017/dbname');  
4  
5  const Posts = mongoose.model('posts', {  
6    _id: mongoose.SchemaTypes.String,  
7    title: mongoose.SchemaTypes.String,  
8    body: mongoose.SchemaTypes.String,  
9    author: mongoose.SchemaTypes.String  
10 });  
11  
12 const posts = await Posts.find();  
13 console.log(posts);
```

References

- Mongoose Quickstart
- Mongoose SchemaTypes
- Model API