Designing the Data Layer with MongoDB and Mongoose

Mark Scott

PLURALSIGHT AUTHOR

@tripletdad99

Code First or Data First?

More is known about how we want the user to interact with the application. Data concerns are secondary or supportive in nature.

We have a better understanding of the business data requirements. Design is secondary or is not as critical to success.

Data First

More specifics about the data

Happy with whatever UI provided

Still needs to meet requirements

We will go with a database first approach

Setting up MongoDB

Demo

Install MongoDB

Studio 3T

Insert sample data

- Users for development
- Sample transactions

Mongoose: The Bridge to MongoDB

Mongoose Courses

Introduction to Mongoose for Node.js and MongoDB

Moving Forward with Mongoose.js

https://app.pluralsight.com/library/search?q=mongoose

Everything in Mongoose starts with a Schema

Schema Example

```
const mongoose = require('mongoose')
const Schema = mongoose.Schema

let userSchema = new Schema({
    email:        String,
    name:         String,
    password:     String,
    createdOn:    Date,
    lastLogin:    Date,
    isActive:     Boolean
})
```

Mongoose Data Types

Mongoose Schema Types	JavaScript Data Types
String	String
Number	Number
Date	Object
Buffer	Object
Boolean	Boolean
Mixed	Object
ObjectId	Object
Array	Array (Object)

Demo

Develop the mongoose schemas

- Users
- Transactions

Stub in the server project

Next Steps

MongoDB Database setup

- Sample transactions added
- Tested in Studio 3T

Mongoose Schemas

- User
- Transaction

Time to work on the API…