**Validations**

Validating new entries for Mongoose models is a breeze. Let's say, for our users we wanted to make the first name, last name, and the email required. We can add our validation in the *server.js* file like this:

*// require the mongoose module*

var mongoose = require('mongoose');

*// to make a model, you can first define a schema, which is just the BLUEPRINT for a model*

var UserSchema = new mongoose.Schema({

first\_name: { type: String, required: true, minlength: 6},

last\_name: { type: String, required: true, maxlength: 20 },

age: { type: Number, min: 1, max: 150 },

email: { type: String, required: true }

}, {timestamps: true });

What you'll notice is that the validation methods in Mongoose are pretty one-dimensional; we pretty much have to validate things using optional parameters in our schema definition. Making required fields is easy, but more intricate validations (email strings, dates, etc) can get ugly. We can validate our data just fine, but the code gets kind of messy. That's a bummer, but there are some things that JavaScript isn't very good for, and this is one of them. To make life a little easier, Mongoose has added a lot of new validation options to help us validate many options.

Take a look at their documentation here:  [Mongoose Validations](http://mongoosejs.com/docs/validation.html" \t "_blank)

**Displaying validation errors:**

Another tiny drawback here: using EJS for errors is not very intuitive. The following code checks to see if our errors are set and print them if they are. Works just fine, but the code isn't very elegant. Again, better tools are on the way with Angular.

**server.js side:**

app.post('/users', function (req, res){

var user = new User(req.body);

user.save(function(err){

if(err){

res.render('index', {errors: user.errors})

}

else {

res.redirect('/users');

}

});

})

**index.ejs side:**

<% if(typeof(errors) != 'undefined' ) { %>

<% for (var x in errors) { %>

<h3><%= errors[x].message %></h3>

<% } %>

<% } %>

**Node Module - Mongoose Validator**

Mongoose's native validation methods may leave something to be desired. If your web application requires a stronger validation engine, there are Node Modules that do just that!

[Mongoose Validator](https://github.com/leepowellcouk/mongoose-validator" \t "_blank) is a simple node module that allows you to create powerful validation arrays that go hand in hand with your models.  Their documentation is super straightforward but please s*pend no more than an hour reviewing it.*