

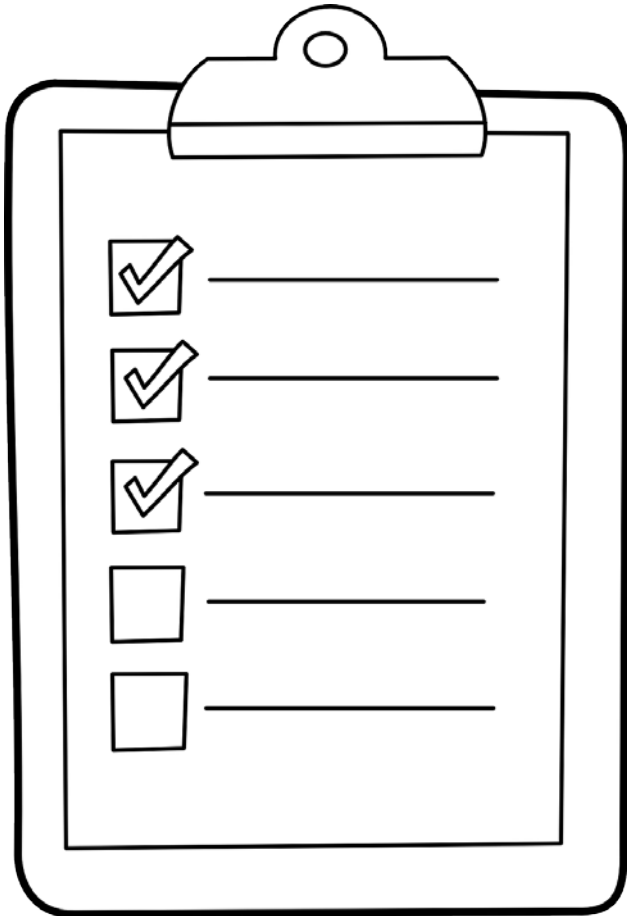
Validation

Mark Scott
[linkedin.com/in/tripletdad99](https://www.linkedin.com/in/tripletdad99)
@tripletdad99



pluralsight 
hardcore dev and IT training

Introduction



Built-in
Validators

Custom
Validators

Handling
Errors

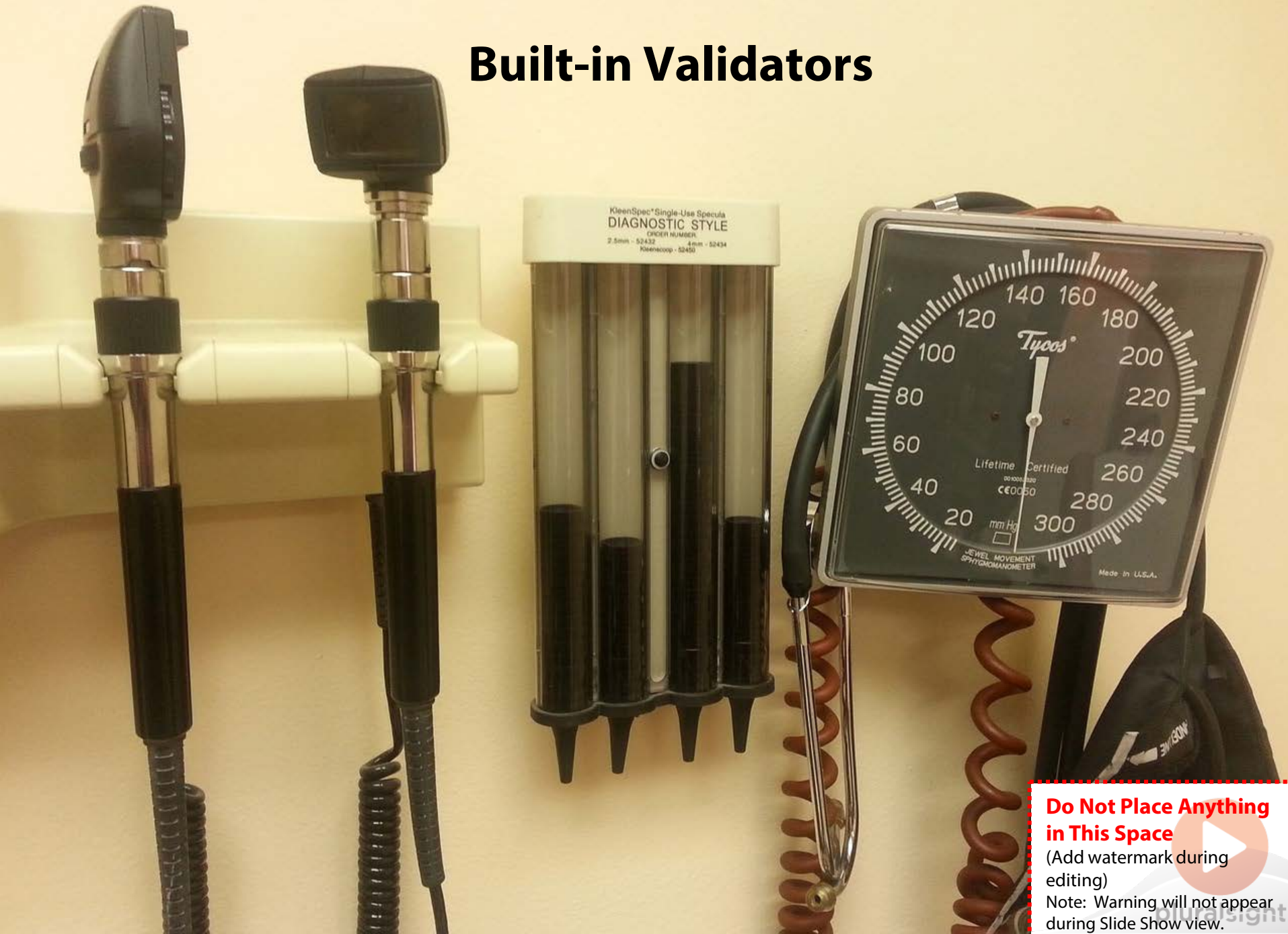
Middleware

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

Built-in Validators



**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

Schema Type	Built-in Validators		
String	required	enum	match
Number	required	min	max
Date	required		
Buffer	required		
Boolean	required		
Mixed	required		
ObjectId	required		
Array	required		

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

Strings - Required

```
// Required Validator Example
var customerSchema = new Schema({
  name:      { type: String, required: true },
  address:   String,
  city:      String,
  state:     String,
  country:   String,
  zipCode:   Number,
  createdOn: Date,
  isActive: Boolean
});

// After the schema is defined - via the path API
customerSchema.path('city').required(true, 'Oops! Supply a city.');
```

// Signature = required(required, [message])

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

Strings – Match and Enum

```
// String - Match Validator Example
```

```
var reMatch = /[a-zA-Z]/;
```

```
var customerSchema = new Schema({  
  name: { type: String,  
          required: true,  
          match: reMatch },
```

```
    // abbreviated...
```

```
});
```

```
// String - Enum Validator Example
```

```
var impediments = ['none', 'minor', 'blocking', 'severe'];
```

```
var standupSchema = new Schema({  
  // abbreviated...  
  impediment: { type: String,  
                required: true,  
                enum: impediments }  
});
```

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

Numbers

```
// Customers must receive at least a 5% discount
var customerSchema = new Schema({
  name:      String,
  // ...
  discount: { type: Number, min: 5 }
});
```

```
// Customers not allowed more than 60% discount
var customerSchema = new Schema({
  name:      String,
  // ...
  discount: { type: Number, max: 60 }
});
```

```
// Customers allowed a discount between 5% and 60% only
var customerSchema = new Schema({
  name:      String,
  // ...
  discount: { type: Number, min: 5, max: 60 }
});
```

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

Middleware

Init

Validate

Save

Remove



**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.


```
// Middleware execution flow example...
var personSchema = new Schema({
  firstName: { type: String, required: true },
  lastName: { type: String, required: true },
  status:    { type: String, required: true, default: 'Alive' }
});

// Build a model from the person schema
var Person = new mongoose.model('Person', personSchema);

// New document instance of a Person model
var newPerson = new Person( { firstName: 'John', lastName: 'Doe' } );

// Save the document... Internal validation (required) kicks off now
newPerson.save(function (err) {
  if (err) return handleError(err);
  // saved the person document!
});
```



**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

Custom Validators



**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.


```
// Custom validation - method signature = validate(obj, [errorMsg])
var sizeValidator = [
  function (val) {
    return (val.length > 0 && val.length <= 50)
  },
  // Custom error text...
  'String must be between 1 and 50 characters long' ];

var personSchema = new Schema({
  firstName: { type: String, required: true, validate: sizeValidator },
  lastName:  { type: String, required: true, validate: sizeValidator },
  status:    { type: String, required: true, default: 'Alive' }
});

// Build a model from the person schema
var Person = new mongoose.model('Person', personSchema);

// New document instance of a Person model
var newPerson = new Person( { firstName: 'John', lastName: 'Doe' } );

// Save the document... And validate
newPerson.save(function (err) {
  if (err) return handleError(err);
  // saved the person document!
});
```

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

Handling Errors



**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.


```
var msg = module.exports = exports = {};  
  
msg.general = {};  
msg.general.default = "Validator failed for path '{PATH}' with value  
'{VALUE}'";  
msg.general.required = "Path '{PATH}' is required.";  
  
msg.Number = {};  
msg.Number.min = "Path '{PATH}' ({VALUE}) is less than minimum allowed  
value ({MIN}).";  
msg.Number.max = "Path '{PATH}' ({VALUE}) is more than maximum allowed  
value ({MAX}).";  
  
msg.String = {};  
msg.String.enum = "'{VALUE}' is not a valid enum value for path  
'{PATH}'.";  
msg.String.match = "Path '{PATH}' is invalid ({VALUE}).";
```

{PATH}

{VALUE}

{TYPE}

{MIN}

{MAX}

```
var mongoose = require('mongoose');  
mongoose.Error.message.Number.min = "{PATH} is too low! Must be at least  
{MIN}.";
```

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

```
// In our controller file...

// New document instance of a Person model
var newPerson = new Person( { firstName: 'John', lastName: 'Doe' } );

// Save the person document now - validation and check for error
newPerson.save(function (err) {
  if (err) {
    console.log('Sorry, there was an error: ' + err);
  }
  else {
    console.log('Your person was saved!');
  }
});

// more real world...
res.render('index', { title: 'Page Title Here', message: errMsg });
```

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

Demo

Validation



**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.

Module Summary

**Built-In
Validators**

**Introduced
Middleware**

**Custom
Validators**

**Handling
Errors**

**Do Not Place Anything
in This Space**

(Add watermark during
editing)

Note: Warning will not appear
during Slide Show view.