

Turn your spaghetti code into ravioli with JavaScript modules

Jeremiah Orr

@JerryOnJava jerryorr.com

Spider Strategies, Inc

thedash.com

What is spaghetti code?



Source: <https://www.flickr.com/photos/avlxyz/4800088303/>

AwesomeConf Registration

Name

Email

Are you presenting?

☒ No

☐ Yes

Credit Card

Security Code / CVV / CVC

Submit Registration!

registration.js

```
$( 'input[name=presenting]' ).change( function () {  
    var presenting = $( 'input[name=presenting]:checked' )  
        .val() == 'yes'  
  
    $( '.payment' ).toggleClass( 'hidden', presenting )  
  
    if ( presenting ) {  
        $( '.payment .has-error' ).removeClass( 'has-error' )  
        $( '.payment .help-block' ).remove()  
        $( '.payment input' ).data( 'valid', true )  
    }  
})
```

registration.js

```
$('#input#name').on('change input blur', function () {  
    var input = $(this)  
    var inputGroup = input.parent()  
  
    inputGroup.find('.help-block').remove()  
  
    var name = input.val()  
    var valid = !empty(name)  
  
    inputGroup.toggleClass('has-error', !valid)  
    if (!valid) {  
        input.after(  
            '<span class="help-block">Name is required</span>'  
        )  
    }  
  
    input.data('valid', valid)  
})
```

registration.js

```
$('#input#email').on('change input blur', function () {  
    var input = $(this)  
    var inputGroup = input.parent()  
  
    inputGroup.find('.help-block').remove()  
  
    var val = input.val()  
    var valid = !empty(val) && /.+@.+\..+/.test(val)  
  
    inputGroup.toggleClass('has-error', !valid)  
    if (!valid) {  
        input.after('<span class="help-block">  
            + 'A valid email address is required</span>')  
    }  
  
    input.data('valid', valid)  
})
```

What is spaghetti code?

~~DRY~~ WET

~~Loosely~~ Tightly Coupled

~~Separation of Concerns~~ Doing Too Many Things At Once

~~Easy to Understand~~ Big Ball of Code

~~Safe to Modify~~ Afraid to Touch

~~Well-tested~~ Tests? Hah!

What is spaghetti code?



Source: <https://www.flickr.com/photos/avlxyz/4800088303/>

What about ravioli?



Source: <http://cocinandoespero.blogspot.com/2009/11/ravioli-rellenos-de-setas-del-bosque.html>

empty.js

```
module.exports = function (v) {  
  if (!v) {  
    return true  
  }  
  
  if (typeof v === 'string' && v.trim() === '') {  
    return true  
  }  
  
  return false  
}
```

required.js

```
var empty = require('./empty')

module.exports = function (val, label) {
  label = label || 'Field'

  return empty(val) ? (label + ' is required') : null
}
```

email.js

```
module.exports = function (val, label) {  
  label = label || 'Field'  
  
  return /.+@.+\..+/.test(val) ? null  
    : (label + ' must be a valid email address')  
}
```

validate.js

```
module.exports = function (validation, label) {  
  return function (value) {  
    return validation(value, label)  
  }  
}
```

validate-input.js

```
var $ = require('jquery')
var _ = require('lodash')

module.exports = function (input, validations) {
  return function () {
    var val = $(input).val()

    var error = _.chain(validations)
      .map(function (validation) {
        return validation(val)
      })
      .compact()
      .first().value()

    return error
  }
}
```

Composed Validation Function

```
var $ = require('jquery')
var required = require('./validation/required')
var email = require('./validation/email')
var validate = require('./validation/validate')
var validateInput = require('./validation/validate-input')

var validateEmail = validateInput(
  $('input#email'),
  [
    validate(required, 'Email'),
    validate(email, 'Email')
  ])

// This will now get the input value, run through
// all the validations, and return an error message
// if validation fails
validateEmail()
```

Even more composition!

watch-and-validate.js

```
module.exports = function (formGroup, validator) {  
  var input = formGroup.find('input')  
  input.on('change input blur', function () {  
  
    formGroup.find('.help-block').remove()  
  
    var error = validator()  
    formGroup.toggleClass('has-error', !!error)  
    if (error) {  
      input.after('<span class="help-block">' +  
        error + '</span>')  
    }  
  })  
}
```


Even more composition!

```
var validateEmail = validateInput(  
  $('input#email'),  
  [  
    validate(required, 'Email'),  
    validate(email, 'Email')  
  ]  
)
```

```
watchAndValidate($('.form-group.email'), validateEmail)
```

```
var validateName = validateInput(  
  $('input#name'),  
  [ validate(required, 'Name') ]  
)
```

```
watchAndValidate($('.form-group.name'), validateName)
```

What is ravioli (modular) code?

DRY

Loosely Coupled

Separation of Concerns

Easy to Understand

Safe to Modify

Well-tested

The UNIX Philosophy

Do one thing and do it well
- Doug McIlroy (1978)

Many UNIX programs do quite trivial things in isolation, but, combined with other programs, become general and useful tools.
- Brian W. Kernighan and Rob Pike (1984)

Small is beautiful.
- Mike Gancarz (1994)

Simple UNIX Tools

- `find` - *searches a directory structure for files matching an expression*
- `xargs` - *executes an arbitrary command using the output of a previous command as arguments*
- `egrep` - *searches for text matching a given regular expression*
- `identify` - *retrieves image metadata (part of ImageMagick)*
- `cut` - *extracts segments of text*
- `tar` - *creates file archives*

A Complex Task

I want to search through my file system for all photos taken during the winter holiday season of every year, and put them in a single archive.

Solved by composing simple programs

```
find . -iname '*.jp*g' -print0 \  
| xargs -0 -L 1 -I @ \  
  identify -format '%[EXIF:DateTime] %d/%f\n' @ \  
| egrep '^[[[:digit:]]{4}:12' \  
| cut -d' ' -f3- \  
| tar -cf december.tar -T -
```

While designing the future, it's
important not to forget the
lessons of the past

*There is such a thing as a Beethoven or Mozart of software design.
- Scott Locklin*

VanillaJS modules

chicken.js

```
(function (global) {  
    var chicken = global.chicken = {}  
  
    chicken.thesis = function () {  
        // I sure hope the browser finished loading egg.js  
        // and sprintf.js by now...  
        return sprintf('You need the %s before the %s.',  
            chicken.sound(), egg.sound())  
    }  
  
    chicken.sound = function () {  
        return 'cluck'  
    }  
  
})(window))
```


VanillaJS modules

egg.js

```
(function (global) {  
    var egg = global.egg = {}  
  
    egg.thesis = function () {  
        // I sure hope the browser finished loading  
        // chicken.js and sprintf.js by now...  
        return sprintf('You need the %s before the %s.',  
            egg.sound(), chicken.sound())  
    }  
  
    egg.sound = function () {  
        return 'crack'  
    }  
  
})(window))
```

VanillaJS modules

debate.js

```
// I sure hope the browser finished loading jquery  
// by now...  
$(document).ready(function () {  
    // I sure hope the browser finished loading egg.js  
    // and chicken.js by now...  
  
    $('<div/>').html(chicken.thesis())  
        .appendTo(document.body)  
  
    $('<div/>').html(egg.thesis())  
        .appendTo(document.body)  
})
```

VanillaJS modules

index.html

```
<script src="jquery-2.1.4.min.js"></script>  
<script src="sprintf.min.js"></script>  
<script src="chicken.js"></script>  
<script src="egg.js"></script>  
<script src="debate.js"></script>
```



browserify

Browserify modules

chicken.js

```
var egg = require('./egg')
var sprintf = require('sprintf')

module.exports.thesis = function () {
  return sprintf('You need the %s before the %s.',
    sound(), egg.sound())
}

var sound = module.exports.sound = function () {
  return 'cluck'
}
```

Browserify and npm

- Over 150,000 packages on npmjs.com
- Can point npm to arbitrary URLs
- Easy to install packages
- Easy to upgrade packages

Is it really that easy?

FORGET THE SLIDES



In just a few minutes, we...





RequireJS

Like Browserify...

- Helps you write tight, focused modules
- Manages complex dependency trees
- Only includes modules you actually use
- Can produce a single bundle

RequireJS

Unlike Browserify (in a good way)

Build not required when developing!

RequireJS modules

chicken.js

```
define(function (require) {  
    var egg = require('./egg')  
    var sprintf = require('./sprintf')  
  
    var module = {}  
    module.thesis = function () {  
        return sprintf('You need the %s before the %s.',  
            sound(), egg.sound())  
    }  
  
    var sound = module.sound = function () {  
        return 'cluck'  
    }  
  
    return module  
})
```

RequireJS modules

index.html

```
<html>  
  <body>  
    <script data-main="scripts/debate"  
      src="scripts/require.js"></script>  
  
</html>
```

RequireJS

Unlike Browserify (in a bad way)

- Cumbersome syntax (IMO)
- No package manager; need to manually install, upgrade, and manage 3rd party library dependences
- More configuration

ES6

ES6 Modules

- ECMAScript version 6 (the next JavaScript standard)
- Native browser support (someday)
- Only loads the modules you need
- Simple syntax

ES6 Modules

chicken.js

```
import * as egg from './egg'
import sprintf from 'sprintf'

export function thesis () {
  return sprintf('You need the %s before the %s.',
    sound(), egg.sound())
}

export function sound () {
  return 'cluck'
}
```

ES6 Modules

- Need something to create a production bundle
- Need a transpiler until supported in target browsers (babel, Browserify + babelify)

Browser compatibility

Desktop		Mobile			
Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari
Basic support	Not supported[1]	Not supported[2]	Not supported	Not supported	Not supported

So which module system should I use?

The nice thing about standards is that you have so many to choose from.

- Andrew S. Tanenbaum

HOW STANDARDS PROLIFERATE:

(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)

SITUATION:
THERE ARE
14 COMPETING
STANDARDS.

14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.



SOON:

SITUATION:
THERE ARE
15 COMPETING
STANDARDS.

Simplicity is prerequisite for reliability.

- Edsger W. Dijkstra

Simplicity is the ultimate sophistication.

- Leonardo da Vinci

Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius - and a lot of courage - to move in the opposite direction.

— Albert Einstein

“Simplicity is about subtracting the obvious and adding the meaningful.”

- John Maeda

Don't write *complex software*...

Write many pieces of *simple, thoroughly tested* software...

And *compose* these simple pieces of software to do complex things.

Thanks for listening!

All code examples available at
<https://github.com/jerryorr/psuweb-modularity>

Jeremiah Orr

@JerryOnJava jerryorr.com

Spider Strategies, Inc
thedash.com