

FUNDAMENTALS OF RESPONSIVE JAVASCRIPT SINGLE PAGE WEB APPLICATIONS



Steve Pember / [@svpember](https://twitter.com/svpember)

JS

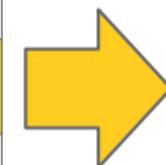
A WEB APPLICATION'S GOAL IS PROVIDE AN ENGAGING USER EXPERIENCE



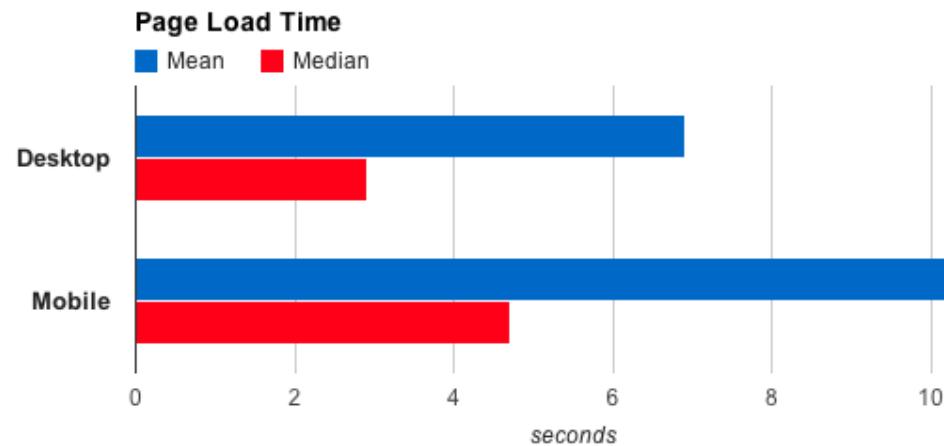
WHAT MAKES AN ENGAGING EXPERIENCE?

- CONTENT
- DESIGN
- PERFORMANCE

Delay	User reaction
0 - 100 ms	Instant
100 - 300 ms	Slight perceptible delay
300 - 1000 ms	Task focus, perceptible delay
1 s+	Mental context switch
10 s+	I'll come back later...



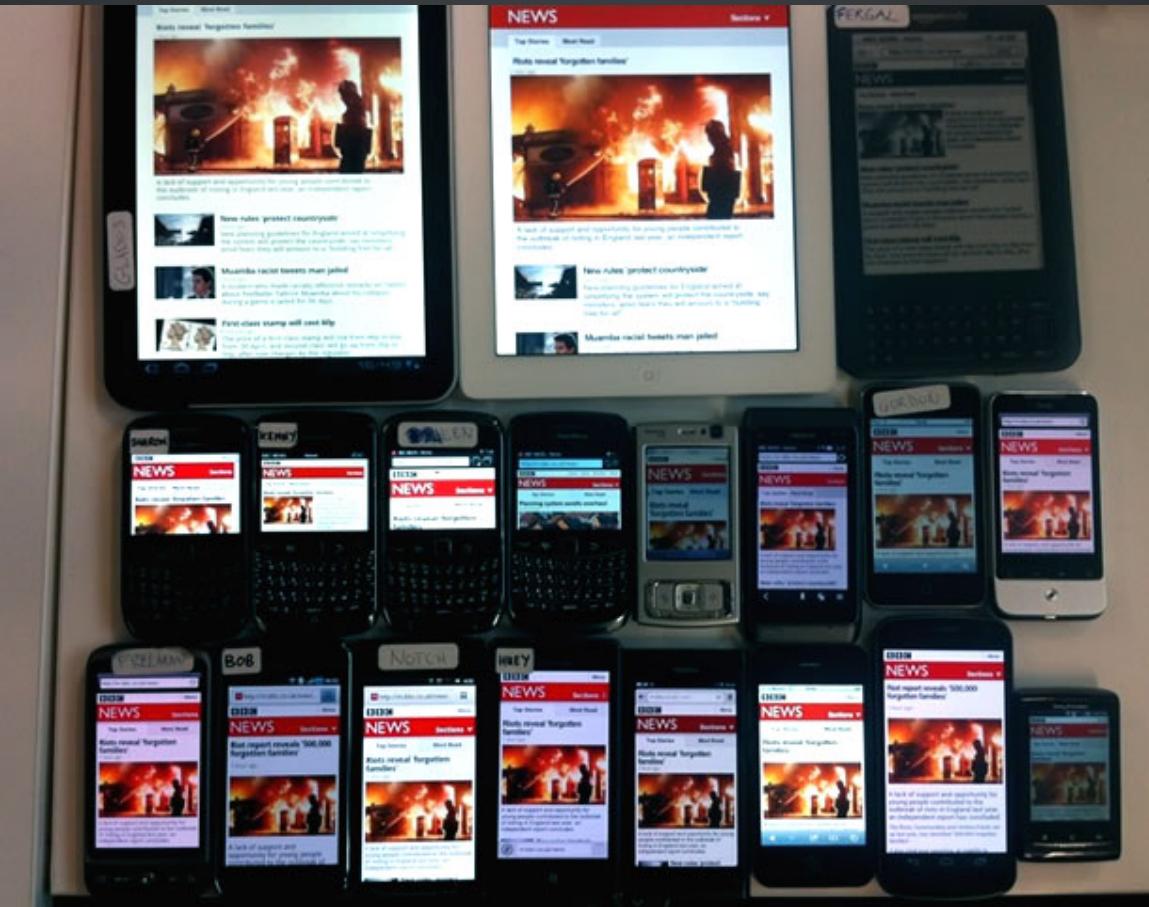
"1000 ms time to glass challenge"



<http://analytics.blogspot.com/2012/04/global-site-speed-overview-how-fast-are.html>

CONTROL WHAT YOU CAN

TRIGGERING A FULL PAGE REFRESH IS HORRIBLE
UX

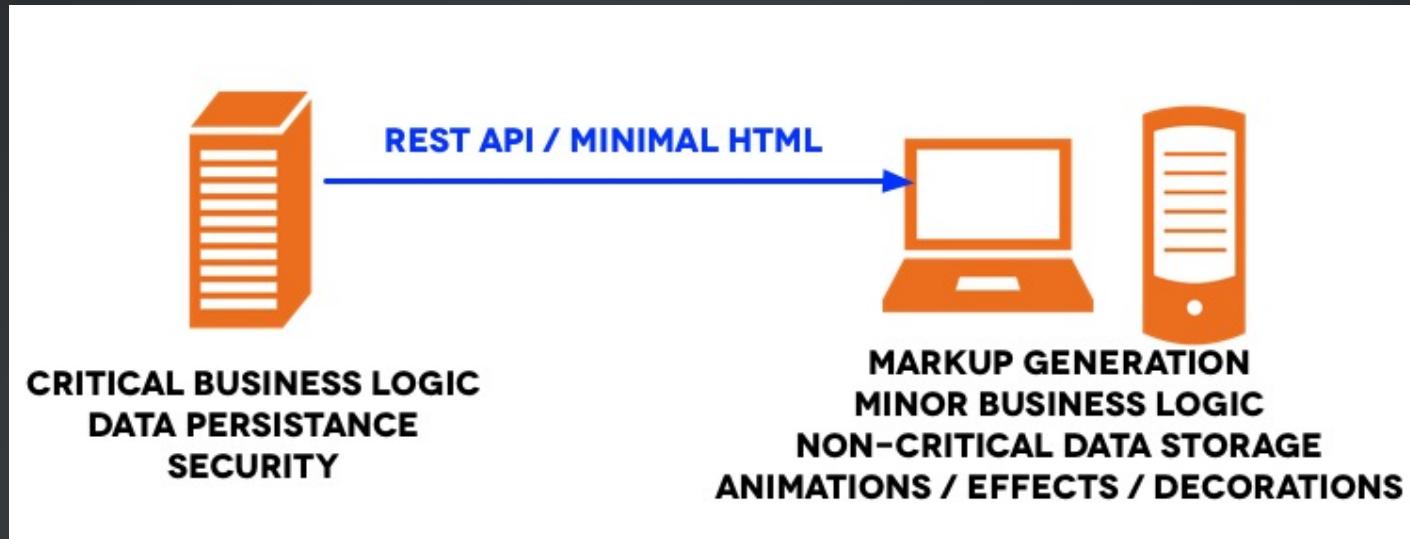


GOAL: UPDATE PAGE CONTENT BASED ON USER INPUT WITHOUT FULL REFRESH

COULD WE JUST USE A SPRINKLING OF AJAX / PJAX?



**YOU SHOULD SPEND AT LEAST AS MUCH
ENGINEERING ON YOUR FRONT END AS YOUR BACK
END**



FINALLY HAVE THE TOOLS

The screenshot shows the Chrome DevTools Audits panel. At the top, there are tabs: Sources, Network, Sources, Timeline, Profiles, Audits, and Console. The Audits tab is selected. Below the tabs is a table with three columns: Self, Total, and Function. The table lists various JavaScript functions and their execution times. A red arrow points from the top-left towards the bottom-right of the table. In the bottom right corner of the table, there is a link labeled "miscellaneous bindings:295".

Self	Total	Function	
0.02%	0.02%	▶ load	content.js:272
0.02%	0.02%	▶ http://www.google-analytics.com/ga.js	content.js:272
0.02%	0.02%	▶ signatures	schemaUtils:50
0.02%	0.05%	▶ forEach	utils:9
0.02%	0.05%	▶ O	end.min.js:29
0.02%	0.02%	▶ Y	end.min.js:52
0.02%	0.02%	▶ (anonymous function)	jquery.min.js:4
0.02%	0.22%	▼ (anonymous function)	ga.js:1
0.02%	0.22%	http://www.google-analytics.com/ga.js	ga.js:33
0.02%	0.02%	▶ Sd	jquery.min.js:3
0.02%	0.02%	▶ h.handle.i	jquery.min.js:295
0.02%	0.02%	▶ sendMessageUpdateArguments	jquery.min.js:2
0.02%	0.03%	▶ (anonymous function)	jquery.min.js:3
0.02%	0.02%	▶ L	jquery.min.js:3
0.02%	0.07%	▶ e.extend.each	jquery.min.js:2
0.02%	0.07%	▶ forEach	utils:9
0.02%	0.07%	▶ elements	elements:1
0.02%	0.07%	▶ Resources	resources:1
0.02%	0.07%	▶ Network	network:1
0.02%	0.07%	▶ Sources	sources:1
0.02%	0.07%	▶ Timeline	timeline:1
0.02%	0.07%	▶ Profiles	profiles:1
0.02%	0.07%	▶ Audits	audits:1
0.02%	0.07%	▶ Console	console:1

Below the table, there are two sections: "Network Utilization" and "Web Page Performance".

- Network Utilization:**
 - ▶ Combine external CSS (5)
 - ▶ Leverage browser caching (15)
 - ▶ Leverage proxy caching (5)
 - ▶ Minimize cookie size
 - ▶ Serve static content from a cookieless domain (8)
 - ▶ Specify image dimensions (3)
- Web Page Performance:**
 - ▶ Optimize the order of styles and scripts (7)
 - ▶ Remove unused CSS rules (933)
 - ▶ Use normal CSS property names instead of vendor-prefixed ones (4)

BONUS: YOUR BUSINESS NOW HAS AN API

(EMBRACE SOA!)

QUICK DEMO!

SINGLE PAGE APPS ARE
THE FUTURE

IF YOU'RE NOT DOING IT, YOU WILL BE LEFT
BEHIND

"Application architectures are changing. The traditional model that the Servlet era APIs were designed for is becoming valid for fewer and fewer cases."

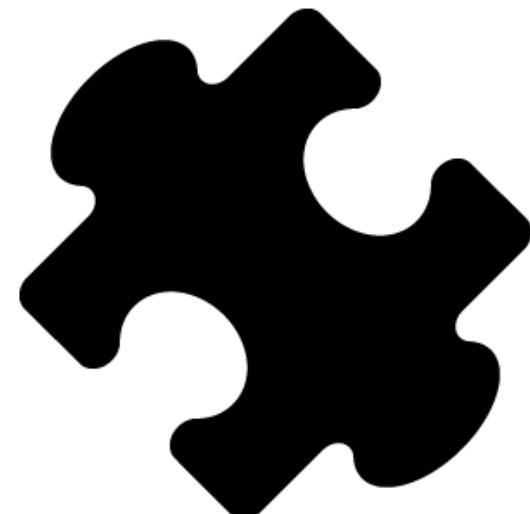
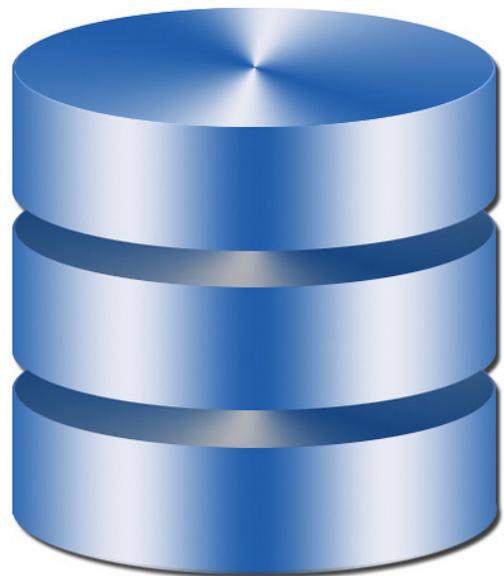
"Mobile and Rich web apps are driving this revolution. Many folks are simply serving up JSON to be consumed by rich clients. So where does Grails fit into this world?"

Graeme Rocher / April 22, 2013

SPA FUNDAMENTALS

?





RESPONSIVE APPLICATIONS

RESPONSIVE DESIGN

- FLEXIBLE GRIDS
- FLEXIBLE IMAGES
- MEDIA QUERIES
- RESPONSIVE ARCHITECTURE
- RESPONSIVE RESPONSES

DYNAMICALLY LOAD / UNLOAD INTERACTIVE
FEATURES BASED ON DEVICE CAPABILITIES

... AKA PROGRESSIVE ENHANCEMENT



TIP: USE MODERNIZR TO RESPONSIVELY ADJUST JS CAPABILITIES

```
var smallBreak = Modernizr.mq("only all and (max-width: 400px)");  
//... later on ...  
if (smallBreak) {  
    // do something for a small screen  
}  
  
//  
// On resize, recalculate the break vars
```

API DESIGN

**THE CLIENT SHOULD ONLY ASK FOR WHAT IT
NEEDS BASED ON DEVICE CAPABILITIES**

(START SMALL AND BUILD UP)

AKA PROGRESSIVE API ENHANCEMENT

```
//  
// A simple example. Build up a map and render it as opposed to a regular DO  
//  
Author author = Author.get(params.id)  
Map json = [ "name": author.name, "id": author.id,  
            bookCount: Book.countByAuthor(author) ]  
if (params.addBooks) {  
    json[ "books" ] = Book.findAllByAuthor(author).collect { book ->  
        book.toJSON() //perhaps a .toJson() method for convenience on a DO?  
        // in Grails 2.3 we'll have custom renderers we could use!  
    }  
}  
  
render json as JSON
```

BANDWIDTH VS LATENCY

(MORE DATA VS MORE REQUESTS)



EXPECT 100-1000MS LATENCY



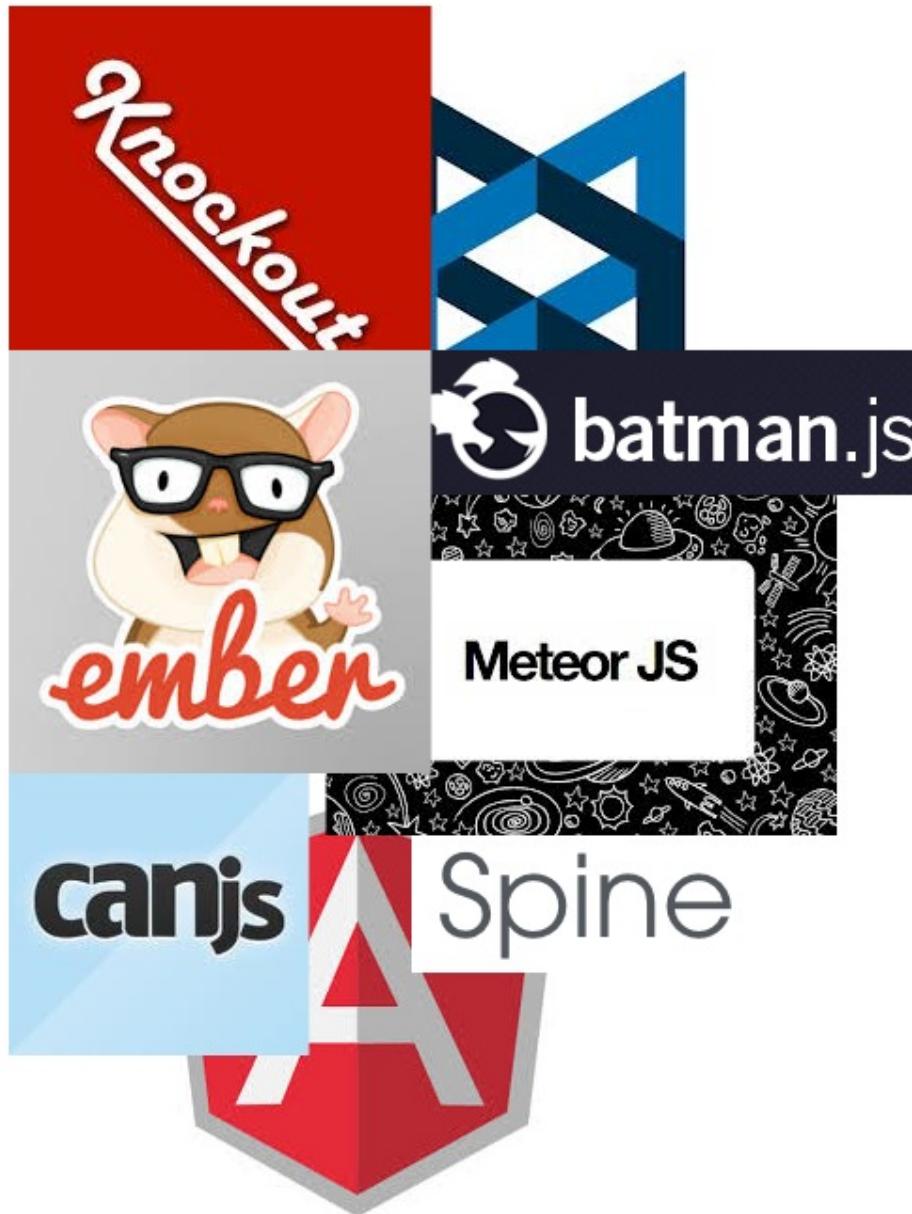
GRAILS 2.3+: REST API

```
import grails.rest.*  
@Resource(uri='/books')  
class Book {  
  
    String title  
  
    static constraints = {  
        title blank:false  
    }  
}
```

BUT RESPONSIVE RESPONSES?

```
class BookController extends RestfulController {  
    static responseFormats = [ 'json' , 'xml' ]  
    BookController() {  
        super(Book)  
    }  
    @Override  
    Book queryForResource(Serializable id) {  
        Book.where {  
            id == id && author.id = params.authorId  
        }.find()  
    }  
}
```

9 ESSENTIAL PROGRAMMING CONCEPTS



SOME ARE MORE RIGID THAN OTHERS



THESE FRAMEWORKS MAY OBFUSCATE THE FOLLOWING:

1. JAVASCRIPT IS WEIRD



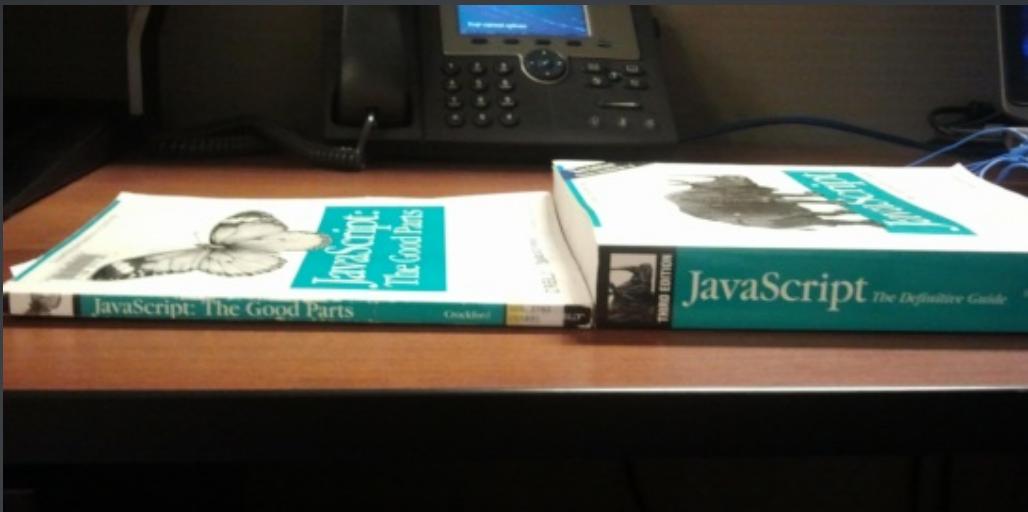
Unearthing the excellence in JavaScript



JavaScript: The Good Parts

O'REILLY® | YAHOO! PRESS

Douglas Crockford



2. PROTOTYPES, NOT CLASSES

AT LEAST UNTIL ECMA 6

```
var Note = Backbone.Models.extend({})  
//  
//    VS.  
  
// use Underscore.js's extend to compose the Backbone Events  
var Note = {};  
_.extend(Note, Backbone.Events);  
// do more!  
_.extend(Note, MyCustomObj);
```

3. EVENTS

ASYNC, EVENT-DRIVEN OR IT'S WRONG

EVENT DELEGATES VS ATTACHED LISTENERS

```
$("table").on("click", "td", function() {
  $(this).toggleClass("chosen");
});
// remember to 'off' when all done!
```

4. REFLOW / REPAINt

WE'RE ALL MAKING VIDEO GAMES NOW

**REFLOW: CHANGES INVOLVING STRUCTURE,
LAYOUT, HEIGHT, ETC.**

**REPAINT: CHANGES INVOLVING VISIBILITY /
VISUALS**

< 16MS

[Open Me in Canary](#)

5. TEMPLATES

```
var source = "<h2>Hello, {{name}}!</h2><h3>Hopefully, you are " +  
    "<span class='focus-word'>{{mood}}</span> this.</h3>";  
  
var template = Handlebars.compile(source); // this step can be done well before  
  
someHttpPromise.then(function(apiData){  
    //sample data: {"name": 'gr8conf', 'mood': 'enjoying'}"  
    var markup = template(apiData);  
    //insert template-generated markup somewhere  
    $("section.present").append(markup);  
});,
```

HELLO, GR8CONF

HOPEFULLY, YOU ARE ENJOYING THIS.

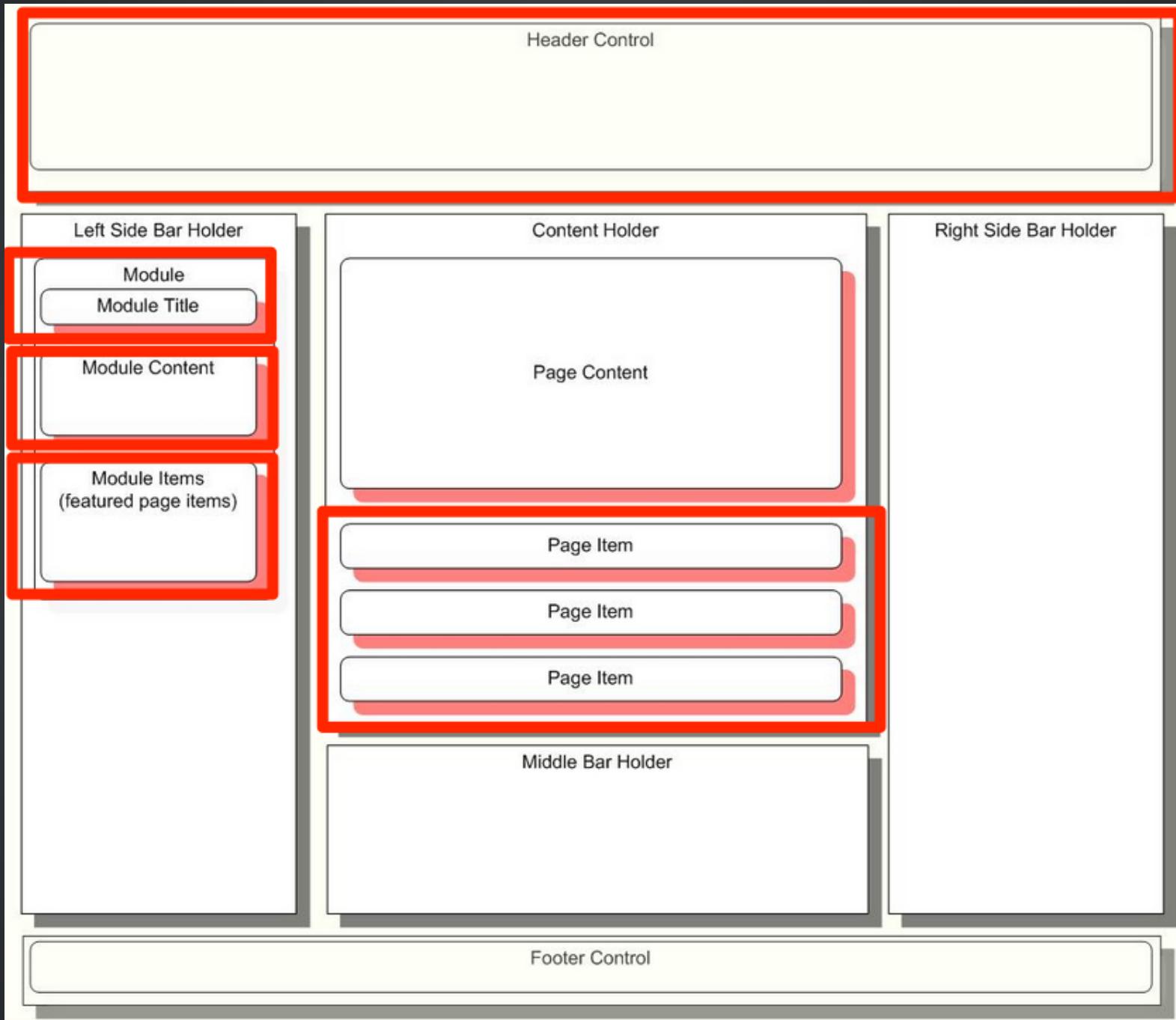
WARNING: TEMPLATE COMPLIATION TIME, AND ...

6. DOM INJECTION

```
var $content = $("#content"),
    data = ["item1", "item2", "item3"],
    pos = data.length;
// 
// Not Great
//
while(pos--) {
    $content.prepend(data[pos]);
}
```

```
var $content = $("#content"),
    $sibling = $content.prev(),
    data = ["item1", "item2", "item3"],
    pos = data.length;
// better
// $content.remove();
while(pos--) {
    $content.prepend(data[pos]);
}
$content.insertAfter($sibling);
```

7. VIEWS / DOM ELEMENT ENCAPSULATION

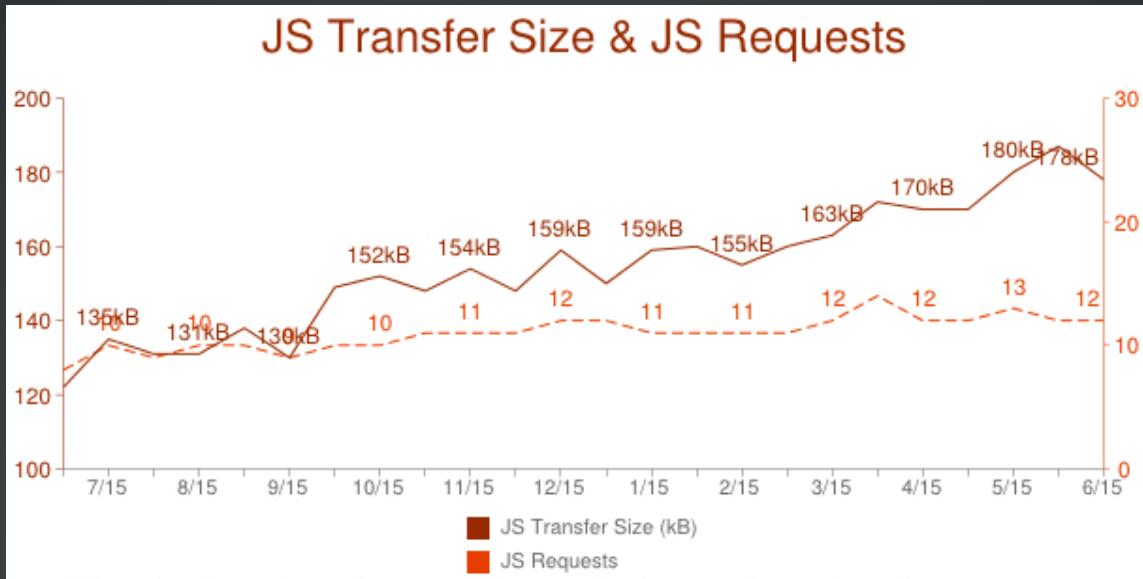


8. ROUTES AND MODULES

JAVASCRIPT WAS DESIGNED TO NOT CARE ABOUT MEMORY USAGE

BUT **YOU** NEED TO

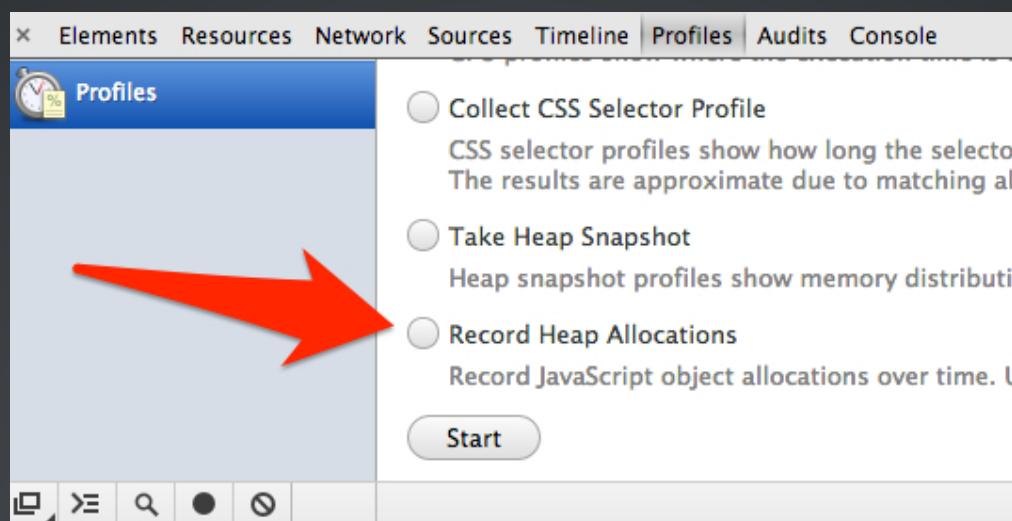




HTTP ARCHIVE: TOP 100 SITES, FIRST VISIT

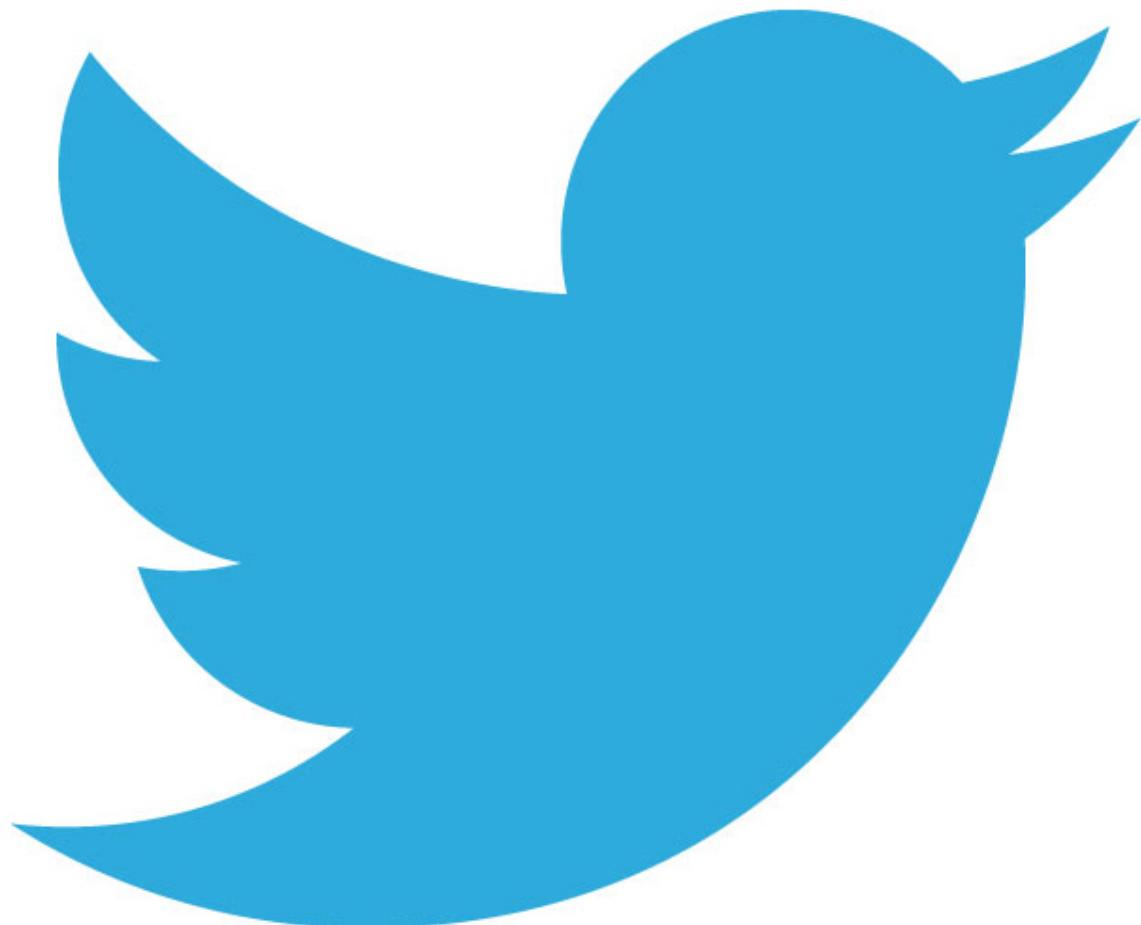
BREAK UP YOUR APP INTO MODULES

**LOAD / UNLOAD MODULES BASED ON CURRENT
ROUTE**



9. MINIMUM VIABLE VIEW

AVOID <BODY></BODY>



SEND USEFUL HTML ON INITIAL REQUEST

AVOID MULTIPLE REQUESTS PER USER ACTION



WHAT ABOUT ROBOTS / SPIDERS?

OH NO, MY SEO!

LIKE US, OUR USERS PLACE A LOT OF VALUE IN SPEED – THAT'S WHY WE'VE DECIDED TO TAKE SITE SPEED INTO ACCOUNT IN OUR SEARCH RANKINGS

-GOOGLE

THE FUTURE OF APPLICATION DEVELOPMENT WILL
BE EXCITING!

JS

THANK YOU!

IMAGE CREDITS

WAT: <http://www.shopify.com/technology/5370262-wat-a-funny-look-at-ruby-and-javascript-oddities>

KING JS: <http://www.hutchhouse.com/blog/who-needs-flash-javascript-is-king/>

1000ms TTG: <http://www.igvita.com/slides/2013/fluent-perfcourse.pdf>

Phone and Tower: <http://www.pcmag.com/article2/0,2817,2405596,00.asp>

Balancing Act: <http://www.govloop.com/profiles/blogs/next-four-years-managing-a-balancing-act>

Yin / Yang: <http://bodydivineyoga.files.wordpress.com/> JS, good parts: <http://oreilly.com/>

Rage/Disapproval Face: <http://knowyourmeme.com/> Colbert Sweat Wipe: <http://www.nofactzone.net/>

Plugin Icon: <http://iconify.it/> Scroll: <http://www.roblox.com/>

