Old Gods & New: A Vision of Backend & Frontend

May 2015

By Gil Tayar
Holy Tantric Markup Language
JavaScript: The Good Parts

Part Six: Ajax Performance

Douglas Crockford
douglas@crockford.com
Paul
Two Tribes
The Backend Tribe
The Frontend Tribe
The Backend Gods
Thread Pools
Message Queues
Scala
And The Frontend Tribe?
Thread Pools
Message Queues
Scala
JavaScript
Angular
DOM
For they were old gods
Serving the Frontend Code
CSS
JavaScript
Please, pray to the backend gods
Thread Pools
Message Queues
Scala
HTML
CSS
JavaScript
LET IT BE SO
HTML, CSS, JavaScript ⇒ FTP
Not Happy
http://mytribe.com/about-us.html
http://mytribe.com/about-us
Please, pray to the backend gods
Thread Pools
Message Queues
Scala
And the backend tribe prayed
LET IT BE SO
URL Rewriting
Secret Sacred Server
Configuration Files
And all was well
Chapter II: Serving ES6 Modules
Smitten with ES6 Modules
An ES6 Module

// aModule.js

export function greet(name) {
  return 'hello ' + name;
}
Importing ES6 Module

```javascript
import {aFunction} from './aModule';

document.onready = function() {
    document.getElementById('greeting').innerHTML = aFunction('anonymous');
}
```
Who transpiles the ES6 to JavaScript?
babel
traceur
esnext
The Sun Will Go Dark

Request
main.js

Response
main.js

Request
sub-a.js

Response
sub-a.js

Request
sub-a-a.js

Response
sub-a-a.js
browserify
webpack
Bundling ES6 Modules
A month later...
The Wondrous F5
(⌘+R for all you Mac-ies)
Bundling Kills Caching

bundle-for-page-1.js
Over-Bundling

bundle-for-page-1.js

bundle-for-page-2.js
Chapter III: The Bold Architecture
Backend Understanding JavaScript
bundles killing caching? over-bundling?

Solved?
HTTP/2
To HTTP/2 Multi-Channel

- Request main.js
  - Response main.js
- Request sub-b.js
  - Response sub-b.js
  - Request sub-a.js
    - Response sub-a.js
  - Request sub-b-a.js
    - Response sub-b-a.js
- Request sub-a-b.js
  - Response sub-a-b.js
HTTP/2 Push Mode

Request
main.js

Response
sub-a.js

Response
sub-b.js

Response
sub-a-b.js

Response
sub-b-a.js
HTTP/2 Push Mode with Cache

Request main.js

Response sub-a.js
Response sub-b.js
Response sub-a-b.js
Response sub-b-a.js
“Please, plead to the gods about JavaScript and HTTP/2”
Thread Pools
Message Queues
Scala
INTRODUCING VENGEANCE! BLAZING CONFRONTATIONS THAT DWARF THE INFINITE!

Read the NEW GODS!

AS THEY REALLY ARE!

Kirby & Thibert
JavaScript
Angular
DOM
Chapter IV: CSS
HTML
JavaScript
Do unto CSS, what you would do unto JavaScript?
Gods That Understand JS Give Us

Auto-transpiling of ES6
Fast serving of JS modules
No cache-killing JS bundles
All using the wondrous F5 key
CSS
LESS
SASS
Compass
Better and Smaller CSS
Yes, CSS has modules

@import 'sub.less';

...
Gods That Understand CSS Give Us

Auto-transpiling of LESS
Fast serving of CSS modules
No cache-killing CSS bundles
All using the wondrous F5 key
“Please, plead to the gods about CSS and HTTP/2”
Thread Pools
Message Queues
Scala
OH, PISS OFF

WE GAVE YOU URL REWRITING ALREADY!
JavaScript
Angular
DOM
Chapter V: HTML and Caching
Browser Caching

Browser always checks if there is a new version
  Last-Modified/Etag
Browser never checks
  Expires: ∞
Caching-Supportive HTML/JS

<script src="main-1.4.js"/>
Caching-Supportive HTML/JS

<script src="main.js">
Caching-Supportive HTML/JS

<script src="main-2014-06-08.js">
“Please, plead to the gods about HTML and infinity caching”
Thread Pools
Message Queues
Scala
OH, PISS OFF

WE GAVE YOU URL REWRITING ALREADY!
Pissed Off!
A Mission from Gods
Chapter VI: Rendering HTML
Single Page Applications
Single Page Applications

Server

Service

page. html

xN times...
Server-side Rendering of HTML
SPA - New Gods Style

Server

Service

page.html

render code

xN times...
SPA - New Gods Style

Server

Service

page.html

render code

xN times...
The backend gods could execute JavaScript?
Sharing Frontend and Backend Code
Thread Pools
Message Queues
Scala
“Please, plead to the gods about CSS and HTTP/2”
OH, PISS OFF

WE GAVE YOU URL REWRITING ALREADY!
JavaScript
Angular
DOM
Message Queues
JavaScript
Hey, Great Idea
LET IT BE SO
- transpile ES6
- minify JS
- serve es6 modules intelligently
- cache es6 modules
- ???

- transpile LESS
- minify CSS
- serve CSS modules intelligently
- cache CSS
- ???

- rewrite links for infinity
- CSS expiring
- serve CSS modules
- HTML server-side
- ???
The Frontend Tribe
Thread Pools
Message Queues
Scala
JavaScript
Angular
DOM
Seriously, though...
Thank You

@giltayar