

# Advanced AngularJS

Siddique Hameed

@siddii

<https://github.com/siddii>

St. Louis JavaScript Meetup - 04/17/2014

# Content Security Policy(CSP)

- Extra layer of security for mitigating Cross-Site Scripting (XSS), data injection attacks, etc.
- *Content-Security-Policy* header restricts fine grain control over contents processed by the browser
- *ng-csp* forbids use of *eval* & *new Function*("")
- *ng-csp* mode runs ~ 30% slower but, it might be a necessary *trade-off* for *certain* kind of applications
- Best suited for privileged app development like Chrome extensions, Chrome packaged apps, Firefox OS etc.
- JavaScript MVC's & Templating frameworks does lots of "*magic*", so be wary of what they are doing under the cover

# CSP...

- Mike West has some excellent articles on CSP to get you started
  - <http://www.html5rocks.com/en/tutorials/security/content-security-policy/>
  - <https://mikewest.org/2012/05/content-security-policy-feature-detection>
- Companies like Github, Twitter have been using it for a while...
  - <https://github.com/blog/1477-content-security-policy>
  - <https://blog.twitter.com/2013/csp-to-the-rescue-leveraging-the-browser-for-security>

# Can I use CSP?

Source: <http://caniuse.com/#search=csp> Dated: 04/10/2014

<a href="#">Show all versions</a>	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	IE Mobile
								2.1		
								2.2		
								2.3		
						3.2		3.0		
						4.0-4.1		4.0		
	8.0		31.0			4.2-4.3		4.1		
	9.0		32.0			5.0-5.1 webkit		4.2-4.3		
	10.0	ms	27.0			6.0-6.1 webkit		7.0		
Current	11.0	ms	28.0	7.0	20.0	7.0	5.0-7.0	4.4	10.0 webkit	10.0 ms
Near future			29.0		21.0					
Farther future			30.0		22.0					
3 versions ahead			31.0							

Notes

Known issues (2)

Resources (3)

Feedback

Edit on GitHub

The HTTP header is 'X-Content-Security-Policy' for Firefox until version 23 and IE10&11, and 'X-Webkit-CSP' for Chrome until version 25 and Safari until version 7. 'Content-Security-Policy' is the official W3C defined header, used by Chrome version 25 and later, Firefox version 23 and later, and Safari 7 and later.

# CSP Demo...

<http://localhost:8000/csp/csp-test.html>

# Mustache Security

## Security Matrix

Spot a mistake? Let us know! We go for fail if unclear - rather too harsh than too lax.

Framework	{SEC-A	{SEC-B	{SEC-C	{SEC-D	{SEC-E	{SEC-F
<a href="#">AngularJS 1.0.8</a>	Fail	Fail	Fail	Fail	PASS	Fail
<a href="#">AngularJS 1.2.0</a>	Fail	PASS	Fail	Fail	PASS	PASS
<a href="#">CanJS</a>	Fail	Fail	PASS	Fail	Fail	Fail
<a href="#">Underscore.js</a>	Fail	Fail	PASS	Fail	Fail	Fail
<a href="#">KnockoutJS</a>	Fail	Fail	Fail	Fail	Fail	Fail
<a href="#">Ember.js</a>	Fail	PASS	PASS	Fail	PASS	TBD
<a href="#">Polymer</a>	TBD	TBD	TBD	TBD	TBD	TBD
<a href="#">jQuery</a>	TBD	TBD	TBD	TBD	PASS	TBD
<a href="#">JsRender</a>	Fail	Fail	Fail	Fail	Fail	Fail
<a href="#">Kendo UI</a>	Fail	Fail	Fail	Fail	Fail	Fail

- **{SEC-A** Are template expressions executed without using eval or Function? (yes = pass)
- **{SEC-B** Is the the execution scope well isolated or sand-boxed? (yes = pass)
- **{SEC-C** Can only script elements serve as template containers? (yes = pass)
- **{SEC-D** Does the framework allow, encourage or even enforce separation of code and content? (yes = pass)
- **{SEC-E** Does the framework maintainer have a security response program? (yes = pass)
- **{SEC-F** Does the Framework allow or encourage safe CSP rules to be used (yes = pass)

Source: <https://code.google.com/p/mustache-security/> Dated: 04/09/2014

# Mustache Security Demo...

<http://localhost:8000/mustache-security/>

# Interpolation

In the [mathematical](#) field of [numerical analysis](#), **interpolation** is a method of constructing new data points within the range of a [discrete set](#) of known data points.

...

*Source: <http://en.wikipedia.org/wiki/Interpolate>*

## **\$interpolate**

Compiles a string with markup into an interpolation function. This service is used by the HTML [\\$compile](#) service for data binding. See [\\$interpolateProvider](#) for configuring the interpolation markup. Compiles a string with markup into an interpolation function.

*Source: [http://docs.angularjs.org/api/ng/service/\\$interpolate](http://docs.angularjs.org/api/ng/service/$interpolate)*



# Interpolation Demo

<http://localhost:8000/interpolate/>

# Transclusion

In **computer science**, **transclusion** is the inclusion of a document or part of a document into another document by reference.

For example, an article about a country might include a chart or a paragraph describing that country's agricultural exports from a different article about agriculture. Rather than copying the included data and storing it in two places, a transclusion embodies **modular design**, by allowing it to be stored only once (and perhaps corrected and updated if the link type supported that) and viewed in different contexts. The reference also serves to link both articles.

...

Source: <http://en.wikipedia.org/wiki/Transclusion>

## ngTransclude

Directive that marks the insertion point for the transcluded DOM of the nearest parent directive that uses transclusion.

Source: <http://docs.angularjs.org/api/ng/directive/ngTransclude>

# Transclusion Demo

<http://localhost:8000/transclude/>

# ng-touch

- Based on touch events from *jQuery mobile*
- Currently, a simple API for *swipe* left, right & clicks
- More enhancements on AngularJS v2.0

# ng-touch Demo

<http://localhost:8000/touch/>

# \$scope *apply* vs *digest*

- Within directives you might likely need *digest* rather than *apply*

Demo: <http://localhost:8000/scope/DigestVsApply.html>

# Using \$injector

- Technique used for debugging, instrumentation etc.
- Integrate external APIs with Angular's DI

```
var ngInjector = angular.injector(['ng']);  
var httpService = ngInjector.get('$http');  
httpService.get('/proxies.json').then(function (response){  
    console.log('Response = ', response);  
});
```

# Meet Mr. Yeoman!



## YEOMAN

- Great tool for rapid scaffolding & prototyping
- If you are already using Grunt/Bower in the workflow, Yeoman can be an easy and valuable addon!



# Angular 2.0 roadmap...

- Designed/targeted for modern browsers supporting ES 6+
- Dirty checking leveraging *Object.observe()* if & where available
- Simplified DI (checkout <https://github.com/angular/di.js>)
- Persistence support for offline mode features
- Better Directives & Templates
- Touch animation support
- Better modularization & Instrumentation support
- Router enhancement

# Q & A?

Code available at [https://github.com/siddii/STLJS\\_04-17-2014](https://github.com/siddii/STLJS_04-17-2014)

*Create github issues if you have more questions or thoughts!*

# References

- <http://angularjs.org>
- <http://www.html5rocks.com/en/tutorials/security/content-security-policy/>
- <http://content-security-policy.com/>
- <https://code.google.com/p/mustache-security/>
- <http://www.slideshare.net/xoomario/jsmvcomfg-to-sternly-look-at-javascript-mvc-and-templating-frameworks>
- <https://developer.chrome.com/extensions/contentSecurityPolicy>
- <https://mikewest.org/2012/05/content-security-policy-feature-detection>
- <http://blog.angularjs.org/2014/03/angular-20.html>
- [https://docs.google.com/document/d/1epha4VgFZVvauFJb2Tx\\_3NJlb3D91PjyZuO5YNAMXoM/edit](https://docs.google.com/document/d/1epha4VgFZVvauFJb2Tx_3NJlb3D91PjyZuO5YNAMXoM/edit)