

AngularConf 2015



71.6M coolshop 21M 92M 20.5M 19.3M 23M 11.2M 5M

Angular vs SEO

Torino, 22 Ottobre 2015
#angularconf15



Chi siamo



Coolshop è una società di ICT solution fondata a Torino nel 2005.

Per seguire meglio i clienti distribuiti su scala europea, nel 2015 abbiamo aperto una sede in Olanda.



Cosa Facciamo

Intranet Applications

Mobile Applications

Web and Ecommerce

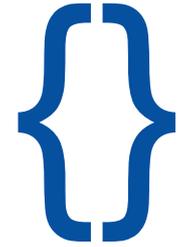
Project Management / Governance

Business Intelligence

Safety Management Systems

Rapid Prototyping

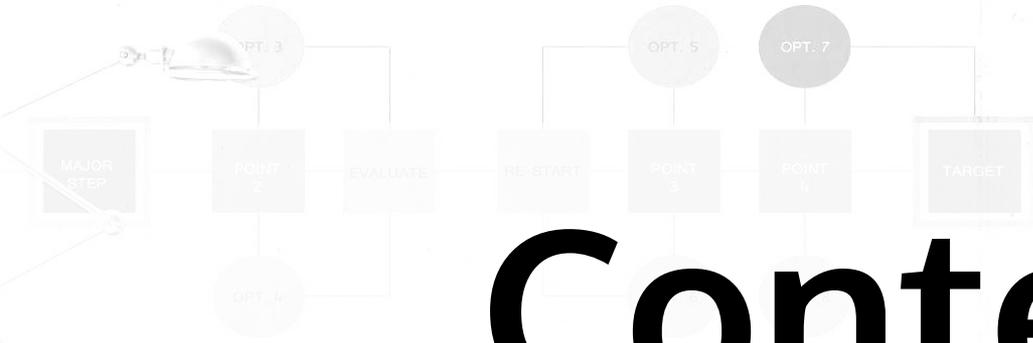
Quality Assurance



AngularConf 2015



70%
82%
36%
51%
97%



Contesto

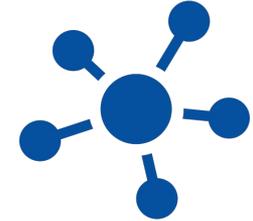


Torino, 22 Ottobre 2015

#angularconf15 | <http://2015.angularconf.it>



Motori di Ricerca



Indicizzano il WEB assegnando dei punteggi alle pagine.

Leggono solo i contenuti testuali
- non eseguono codice lato client, js.

Crawler e Spider software in grado di leggere i contenuti e navigare attraverso i link trovati.

Più è alto il punteggio ottenuto, più visibilità si avrà.



SEO (Search Engine Optimization)

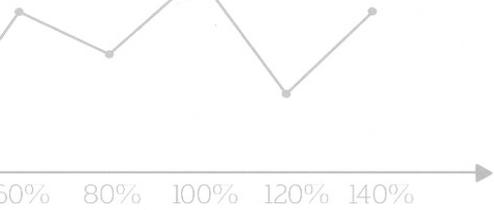
SEO importante per visibilità e marketing.
In generale tutte le attività volte a migliorare il ranking sui motori di ricerca.



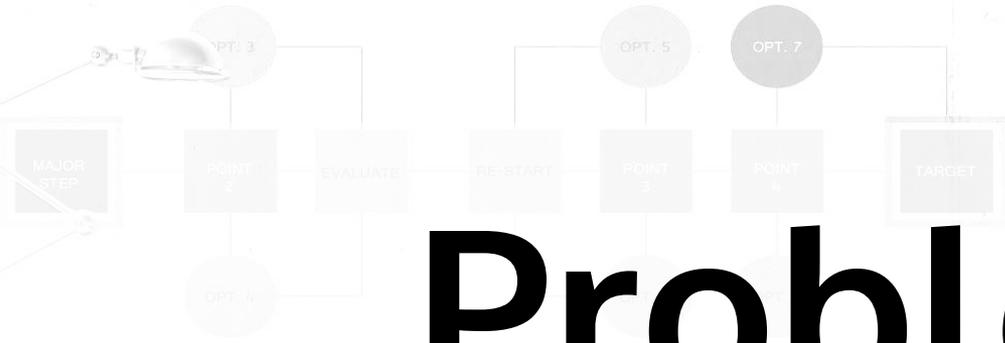
Definisce delle best practice per migliorare il punteggio assegnato alle pagine

- metatag specifici e diversi per ogni pagina
- evitare ridondanza dei contenuti
- HTML semantico

AngularConf 2015



- 70%
- 82%
- 36%
- 51%
- 97%



Problema





Il problema delle SINGLE PAGE APPLICATION

Framework Javascript come AngularJs sono sempre più diffusi.

Molta logica viene spostata lato client (MVC, Routing)

I contenuti dinamici vengono iniettati tramite Javascript (AJAX), quindi **invisibile** ai motori di ricerca



Il problema delle SINGLE PAGE APPLICATION

Google is reading your templates, not your data

When Google indexes your pages, it sees

```
<h1>{{appName}} - {{pageName}} - {{username}}</h1>
```

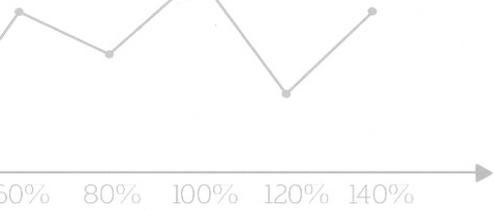
or maybe just

```
<h1></h1>
```

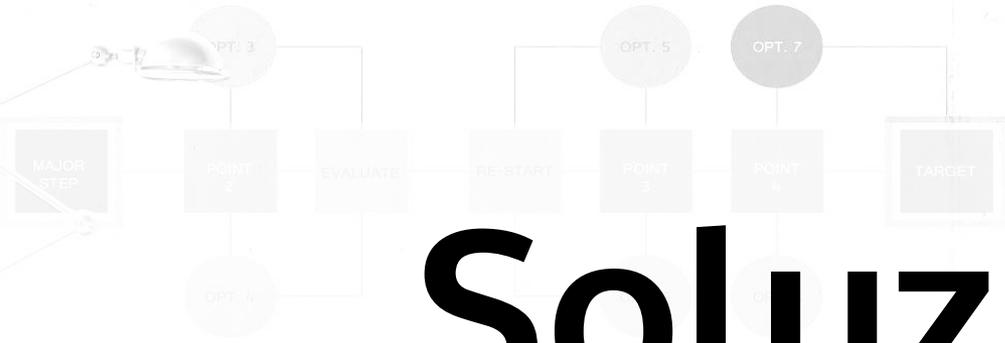
but you need it to see

```
<h1>Awesome App - User Profile - Fred James</h1>
```

AngularConf 2015



70%
82%
36%
51%
97%



Soluzione





Rendiamole indicizzabili



- Differenziare risposte in base al chiamante
- Ajax Crawling Scheme
- Regole di rewrite basate su user agent
- PRECOMPOSED APPs (solo se spider esegue js)



Ajax Crawling Scheme



Un accordo tra crawler e server; si rende noto al crawler che viene rispettato lo schema con

- Hashbang URL
- apposito metatag

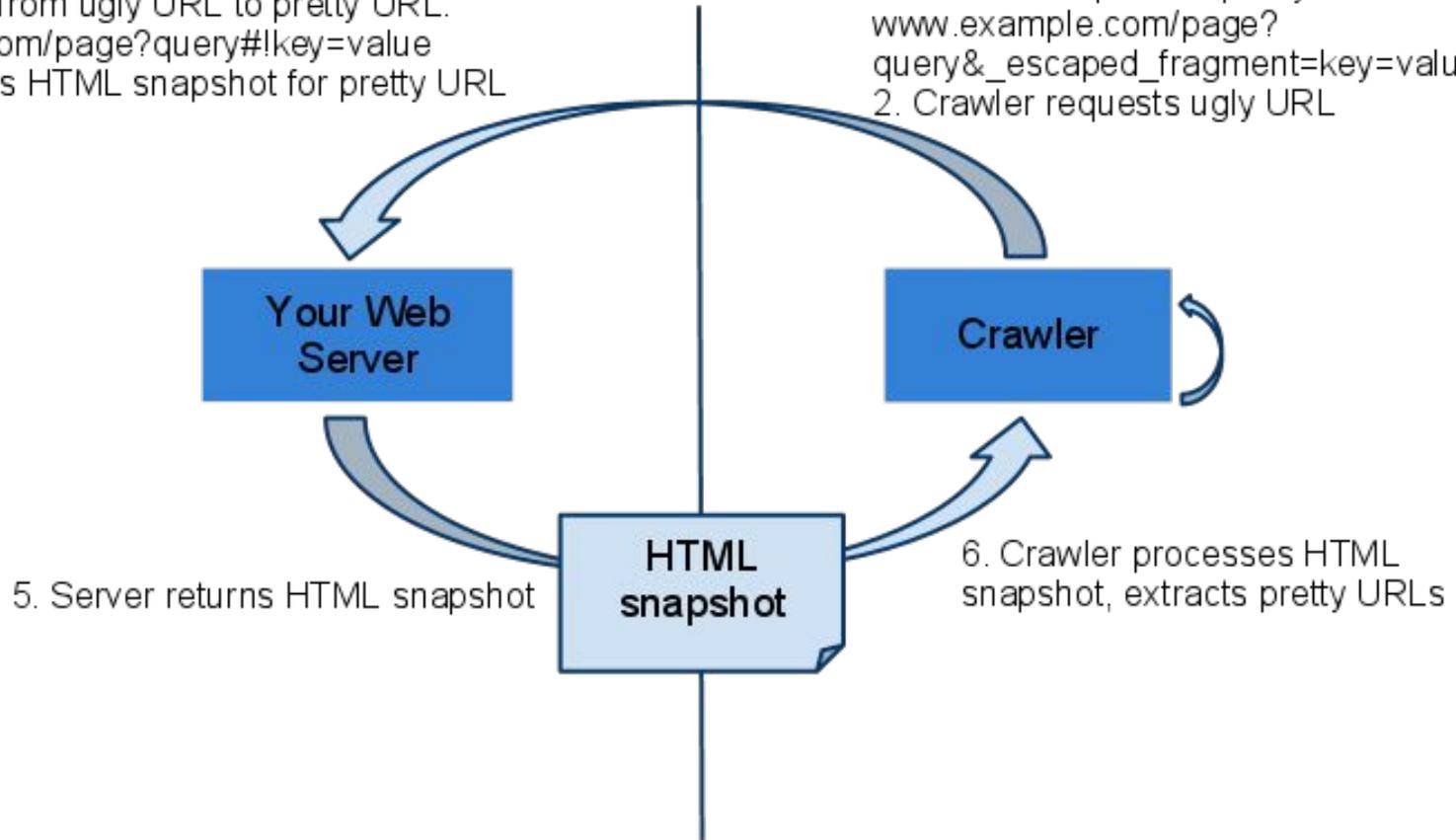
Il crawler inoltra una richiesta utilizzando il parametro
`_escaped_fragment_`

Il server può diversificare le risposte in base al chiamante (utente o crawler)



- 3. Server maps from ugly URL to pretty URL:
`www.example.com/page?query#!key=value`
- 4. Server creates HTML snapshot for pretty URL

- 1. Crawler maps from pretty URL to ugly URL:
`www.example.com/page?query&_escaped_fragment=key=value`
- 2. Crawler requests ugly URL



5. Server returns HTML snapshot

6. Crawler processes HTML snapshot, extracts pretty URLs



Bello ma chi lo supporta?

Tutti i maggiori motori di ricerca

SEARCH ENGINE

AJAX CRAWLING SCHEME

Google

SI

bing

SI

Yandex

SI



Bello ma chi lo supporta?

E alcuni social network

SOCIAL NETWORK



AJAX CRAWLING SCHEME

SI

SOLO CON HASH BANG (#!)

NO

NO

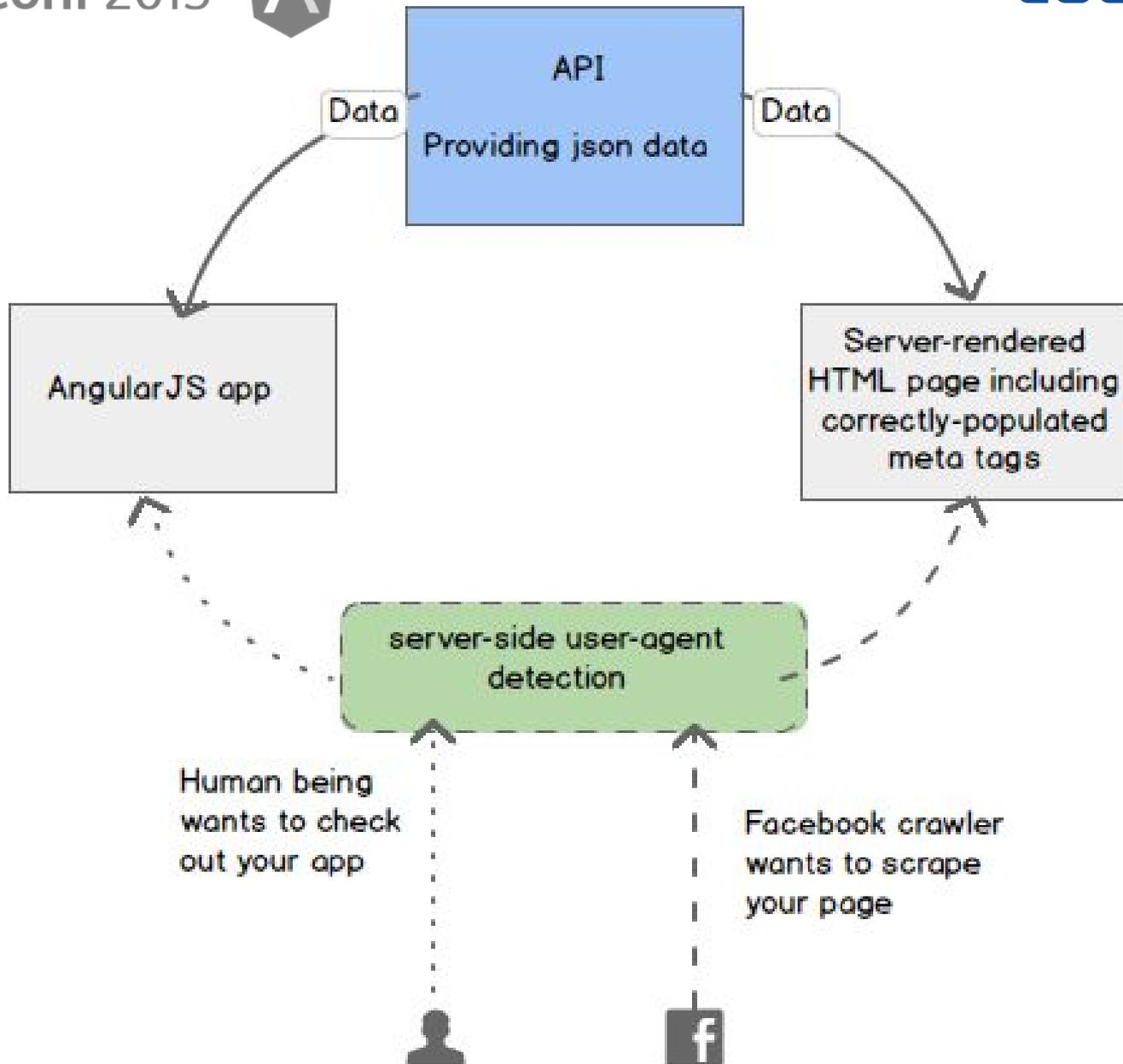


Regole di rewrite basate su user agent



Come nell'ajax crawling schema vengono diversificate le risposte in base al chiamante, ma senza nessun "accordo" con il crawler.

La diversificazione viene fatta basandosi principalmente sullo user agent tramite espressioni regolari.





HTML Snapshot: configurazione server

URL Rewrite e Remapping



- riconoscere una particolare situazione a partire dalla URL richiesta
- decidere in base allo user agent o presenza di parametri, quale risposta servire

Tecnica possibile con tutte le tecnologie Server (Apache, Nginx, Tomcat, Express), ogni ambiente richiede la propria configurazione.



HTML Snapshot: configurazione server



Esempio Apache:

```
RewriteEngine On
```

```
RewriteCond %{QUERY_STRING} _escaped_fragment_
```

```
RewriteRule ^(.*) /snapshots/$2 [P,L]
```



Generazione Snapshot



Uso di headless browser capaci di eseguire Javascript (Phantom, Zombie)

- A. Generazione asincrona
- B. Generazione sincrona (runtime)
- C. Terze parti



Generazione Snapshot



A: Generazione asincrona:

- Html snapshot pregenerati. Risposte salvate in file HTML statici, da servire ai crawler
- La creazione degli snapshots può essere automatizzata. (Ant, Grunt, Cron, Scheduled jobs)
- Vantaggi scalabilità, velocità
- Svantaggi aggiornamento snapshot, complessità



Generazione Snapshot



B: Generazione sincrona (runtime)

- Ad ogni richiesta viene creato uno snapshot (non viene salvato nessun html statico) da servire ai crawler.
- Vantaggi aggiornamento snapshot, complessità
- Svantaggi scalabilità, velocità



Generazione Snapshot



C: Terze parti

- Appoggiarsi a servizi di terze parti per la generazione degli snapshots.
- Vantaggi nessun setup, affidabilità
- Svantaggi poco personalizzabile, affidabilità



Configurazione applicazione angularJs

Dichiariamo di rispettare l'ajax crawling schema:

- utilizzando le Hashbang URL
- inserendo nell'header l'apposito metatag



Configurazione applicazione angularJs

```
<head>
```

```
<!-- ... -->
```

```
<!-- Dichiarare l'URL base dell'applicazione -->
```

```
<base href="/">
```

```
<!-- Comunicare al crawler che viene rispettato lo schema AJAX CRAWLING -->
```

```
<meta name="fragment" content="!">
```

```
<!-- ... -->
```

```
</head>
```

```
var app = angular.module('app')
```

```
.config(['$locationProvider', function ($locationProvider) {
```

```
  // ...
```

```
  // HTML5 History API - browser moderni
```

```
  $locationProvider.html5Mode(true);
```

```
  // or this use for old browser compatibility
```

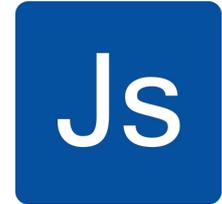
```
  $locationProvider.hasPrefix("!");
```

```
});
```



Googlebot esegue Javascript

Spider di nuova generazione, in grado di eseguire Javascript, seppur con qualche limite:

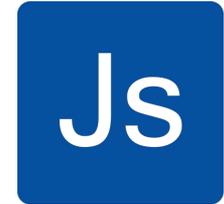


GOOGLEBOT **NON** ESEGUE JAVASCRIPT SE:

- il codice javascript viene eseguito dopo una chiamata AJAX o un timeout



Googlebot esegue Javascript



GOOGLEBOT **ESEGUE** JAVASCRIPT SE:

- uno script è caricato mediante HTML TAG `<script>` ed eseguito immediatamente dopo essere stato caricato
- una funzione è delimitata ed eseguita allo scatenarsi dell'evento "DOMContentLoaded" o dell'evento "window.onload"



Precomposed APPs

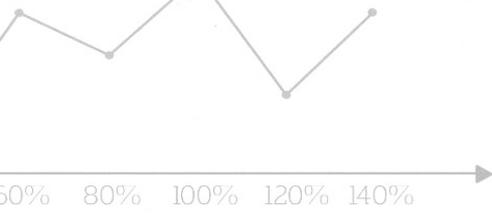
Un'alternativa possibile grazie a crawler capaci di eseguire codice lato client.

Le risorse (HTML partials) e i **dati** che necessitano di caricamento asincrono sono già presenti in pagina.

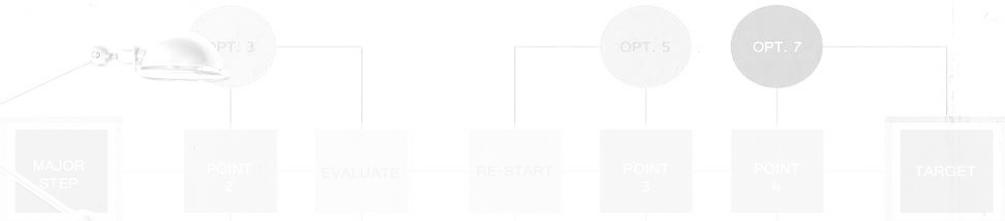
html2js + \$templateCache:

- ottimizzazione delle prestazioni dell'applicazione
- riduzione del traffico di rete
- HTML partials caricati mediante tag `<script>` e quindi disponibili all'esecuzione da parte di Googlebot

AngularConf 2015



70%
82%
36%
51%
97%



Best Practice





AngularJs and SEO best practice



Visualizzare metatags specifici per ogni pagina

- si può creare un proprio servizio per iniettare i metatag nelle view
- oppure appoggiarsi a direttive esistenti come 'viewhead'

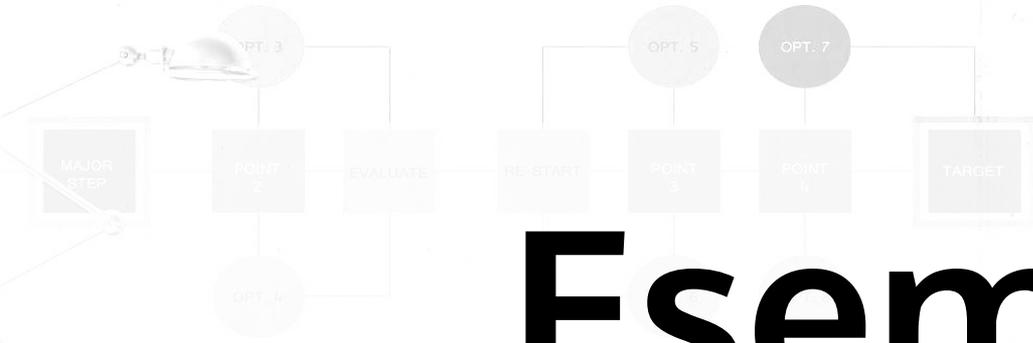
Sitemap

- può essere generata con l'ausilio di task grunt o simili

AngularConf 2015



70%
82%
36%
51%
97%



Esempio

<https://github.com/Coolshop/angular-seo-demo>



AngularConf 2015



coolshop

Grazie

alessandro@coolshop.it

www.coolshop.it

Torino, 22 Ottobre 2015
#angularconf15