Google, Javascript et Ajax : limites techniques et réalité actuelle (2e partie : Angular JS)

Par Philippe Yonnet

Domaine :	Recherche	**Référencement**
Niveau :	Pour tous	**Avancé**

De plus en plus de sites web sont réalisés en utilisant des technologies comme Javascript ou l'Ajax. Pour les explorer, les moteurs de recherche doivent s'adapter quotidiennement à cette nouvelle donne. Mais arrivent-ils réellement à crawler et analyser aujourd'hui tous les sites, parfois complexes, qu'ils explorent ? Cet article en deux parties a pour objectif de faire un point impartial sur la réalité actuelle et les limites, obligatoires, que l'on trouve actuellement dans les possibilités des robots d'aujourd'hui. Le mois dernier, nous avons expliqué les limites de l'Ajax. Ce mois-ci, nous étudions le fonctionnement et évoquons la compatibilité SEO de frameworks Javascript comme Angular JS.

Dans l'article du mois dernier, nous nous étions intéressés aux méthodes disponibles pour rendre l'Ajax crawlable par les robots des moteurs. Mais il existe également d'autres méthodes pour créer des pages web à l'aide de frameworks Javascript comme Angular JS, Backbone JS, Knockout JS ou Ember JS. L'utilisation de ces techniques pose le même type de problème pour le référencement : sans précautions, l'emploi de ces outils rend le contenu des pages web impossible à faire explorer par les moteurs.

Pourtant, ces méthodes deviennent très populaires chez les développeurs web, car elles simplifient leur travail en le rapprochant du développement d'une application (ces frameworks sont faits pour fabriquer des SPA : des « *single page applications* »). Quelle position adopter dans ce cas ? Peut-on rendre explorable par Googlebot un site web développé comme une SPA à l'aide d'un framework JS ? Peut-on en attendre un bon référencement ? Si non, faut-il bannir l'emploi de ces outils ?

Nous allons essayer de répondre à ces questions dans cette deuxième partie, en nous intéressant au plus populaire de ces frameworks Javascript : Angular JS.

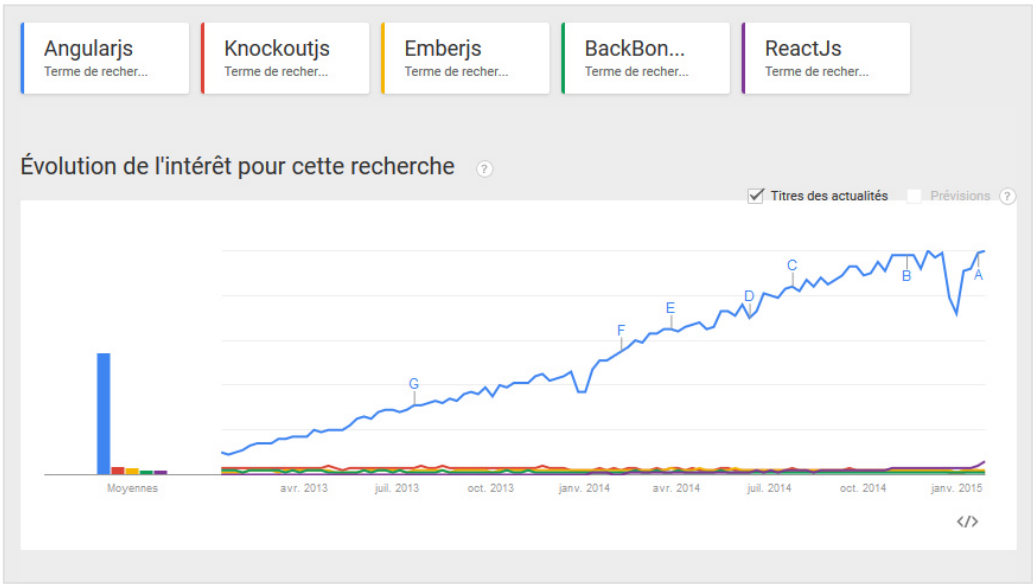

Fig. 1. Un rapide coup d'œil sur Google Trends permet de se rendre compte que la popularité du framework JS ne cesse d'augmenter. Les offres d'emploi pour des développeurs maitrisant cet outil ont explosé en 2015.

Angular JS : le framework développé avec le soutien de Google

AngularJS est un projet qui a été lancé par Misko Hevery, un employé de Google, dont la principale mission est d'être un coach en méthodes Agile pour les développeurs de la firme de Mountain View. Il est à l'origine de plusieurs projets Open Source, dont AngularJS.

Le framework reçoit le soutien actif de Google, dont les ressources en communication, marketing, et la notoriété ne sont pas pour rien dans le succès de l'outil chez les développeurs.

On pourrait penser qu'un outil développé par Google serait « SEO compliant » ? Ce n'est pas si simple… Au point qu'on sent parfois un peu d'embarras des Googlers quand on leur pose des questions sur l'emploi de ces outils sur un site web : impossible de déconseiller d'employer une technologie développée avec le support de Google. Mais impossible également de vraiment la conseiller à cause … de ce que nous allons vous expliquer maintenant.

Premier contact du SEO avec AngularJS : où est passé le contenu ?

Si vous testez un site fait avec Angular JS avec un crawler fait pour le SEO, comme Screaming Frog par exemple, le résultat sera sans appel :

1. Le crawler risque fort de s'arrêter de crawler après le téléchargement de la première page …
2. Et de plus, un coup d'œil sur cette page révélera que cette page est vide de contenu !
Bref, apparemment, le site ne peut pas être SEO friendly.

Mais ce résultat est logique car :
• Le code HTML renvoyé lors de l'appel de la Home Page est un template AngularJS, dont la mission est de permettre à votre browser de télécharger le code Javascript qui, lui, va générer le contenu HTML dynamiquement, mais dans votre navigateur.
• Bref, comme un crawler de type Screaming Frog ne sait pas exécuter du Javascript, il ne peut pas accéder au contenu.

La tentation est grande, dans ces conditions, de décréter que **le site est totalement incompatible avec un bon référencement.**
Sauf que c'est aller un peu vite en besogne.

Regardons un peu sous le capot avec un example : le site jobfoundry.com

Pour un utilisateur, sa page d'accueil ressemble à ceci :

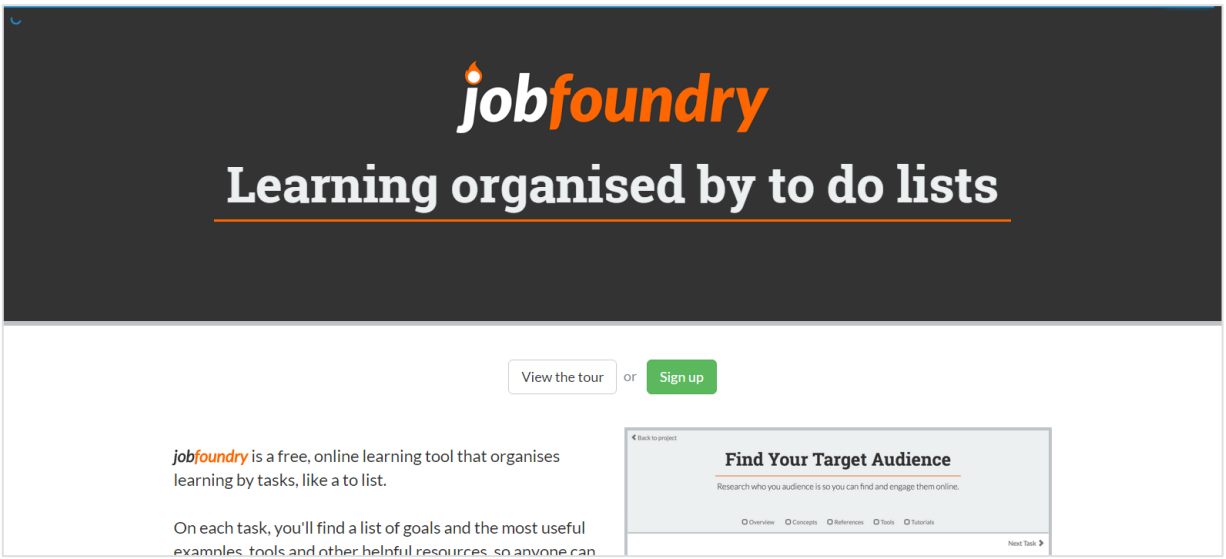

Fig.2. La page d'accueil du site jobfoundry.com

Pour votre crawler SEO habituel, la page d'accueil ressemble à la figure 3 (la page dans un navigateur avec Javascript désactivé).

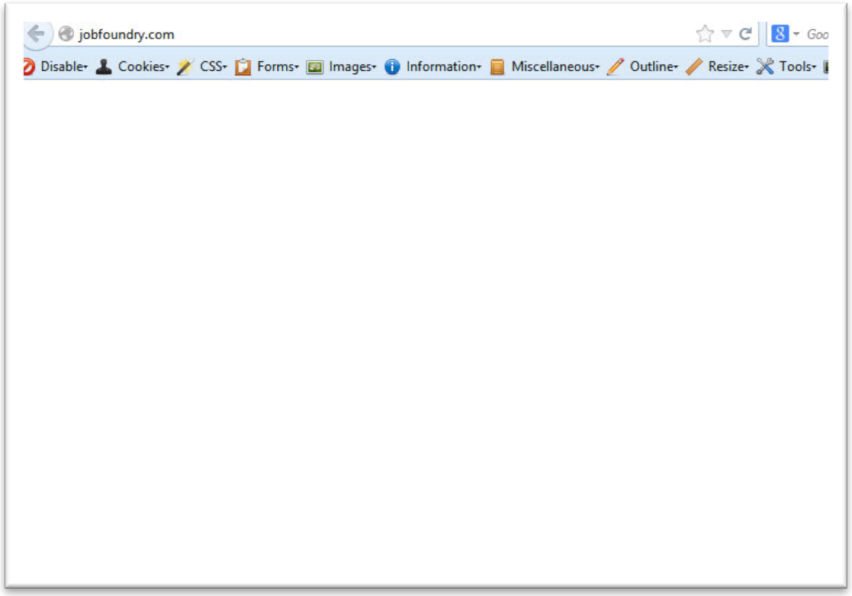

Fig.3. La page d'accueil du site jobfoundry.com vue par un spider « classique »

Parce qu'en réalité, le code source de la page est celui-ci :

```
<!DOCTYPE html>
<!--[if lt IE 7]> <html class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->
<!--[if IE 7]> <html class="no-js lt-ie9 lt-ie8"> <![endif]-->
<!--[if IE 8]> <html class="no-js lt-ie9"> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js"> <!--<![endif]-->
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<meta name="fragment" content="!">
<title>Jobfoundry - learn fast and build things</title>
<base href="/">
<meta name="description" content="">
<meta name="viewport" content="width=device-width">
<link
href='http://fonts.googleapis.com/css?family=Lato:300,400,300italic,400italic|Merriweather:400,
300|Roboto+Slab:400,700' rel='stylesheet' type='text/css'>
<!-- build:css({app,.tmp}) /css/main.css -->
<link rel="stylesheet" type="text/css" href="/css/vendor.css">
<link rel="stylesheet" type="text/css" href="/css/app.css">
<!-- endbuild -->
</head>
<body ng-app="jobFoundryApp">
<!--[if IE]>
<script src="//html5shiv.googlecode.com/svn/trunk/html5.js" rel="dns-prefetch" ></script>
<![endif]-->

        <!-- Add your site or application content here -->
<div ng-view></div>
```

```
<ng-toast></ng-toast>
<script>
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','//www.google-analytics.com/analytics.js','ga');

ga('create', 'UA-42663789-1', 'jobfoundry.com');
</script>
<script src="/js/vendor.js"></script>
<script src="/js/app.js"></script>
<script type="text/javascript">
window.user = null;
</script>
</body>
</html>
```

Bref, le code source contient les appels au code, mais c'est tout !

Google a-t'il accès au contenu de ce site : OUI, car Google sait exécuter du Javascript. Google peut-il facilement explorer et indexer des contenus faits de cette façon : oui, mais à condition de vraiment lui faciliter la vie.

Ce que Google indexe par défaut sur les sites en AngularJS

Sans précautions particulières, Google va indexer les templates AngularJS et c'est tout. On reconnait facilement ces cas dans les pages de résultats de Google car le snippet ressemble à celui de la figure 4.

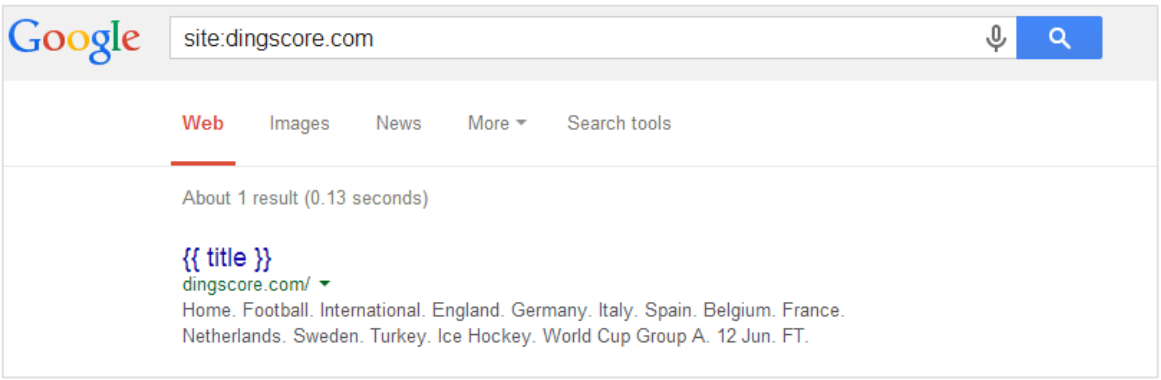

Fig. 4. Exemple de snippet affiché par Google pour un site web (mal) réalisé en AngularJS.

La variable entre crochets est censée être remplacée dynamiquement par une valeur de balise Title dans le navigateur, mais Google indexe la variable avec sa syntaxe Angular, pas la valeur.

Comment faire en sorte que le contenu soit entièrement indexé ?

Pour que les sites AngularJS soient entièrement indexés, les méthodes proposées pour l'Ajax sont valides. Avant que Google déclare obsolète la méthode, les développeurs avertis utilisaient la méthode du hashbang (autrement appelée la méthode des « escaped fragments »). Dorénavant, la méthode à base de « pushState() » est recommandée.

Cependant, utiliser ces approches est nécessaire, mais souvent insuffisant… Car Google rencontre toujours d'énormes difficultés à explorer et indexer correctement et intégralement les contenus générés en Javascript.

Dans tous les cas, il sera donc nécessaire de **tester** si un site fait en AngularJS est réellement explorable avant de le mettre en production.

Comment tester si un site est réellement explorable par Google

Pour vérifier si le contenu est accessible à Googlebot, la première chose à faire est de tester la rendition de la page avec l'outil « fetch as Googlebot » dans la Search Console. Si la page affichée dans le « snapshot » est conforme à la page que vous attendez, cela signifie que le moteur de rendition de Googlebot est capable d'accéder au contenu de votre page (voir figure 5).

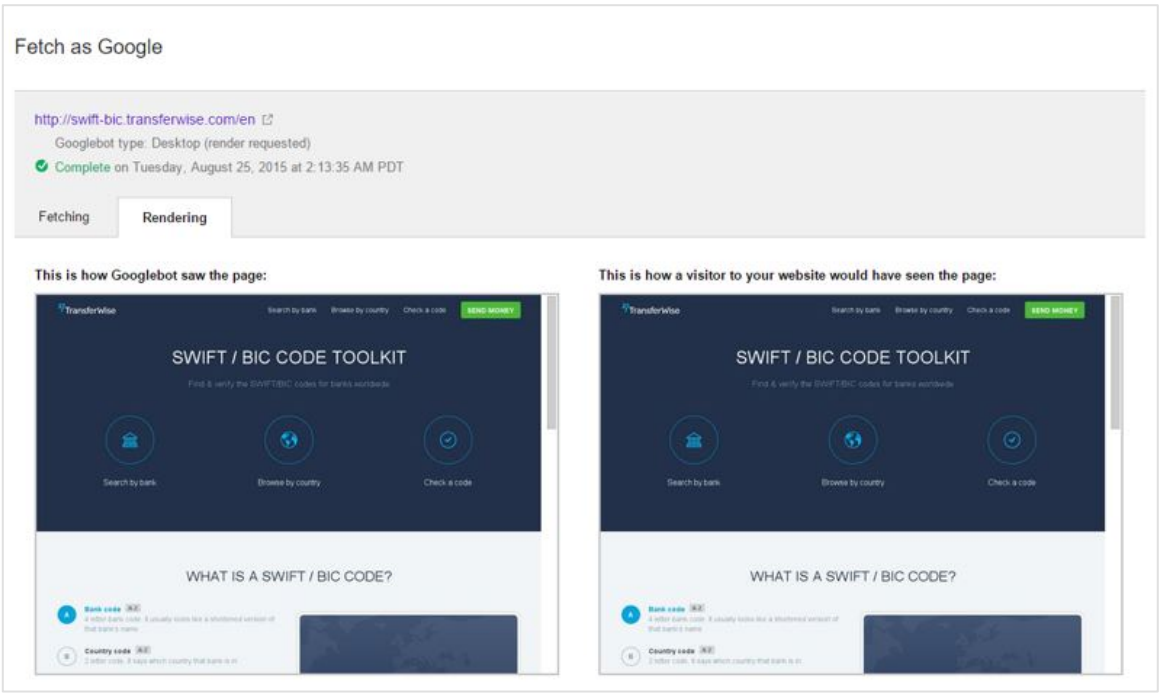

Fig. 5. Exemple d'une rendition avec « Fetch as Google » testée par l'équipe de Deepcrawl sur un site en AngularJS : le résultat est bon.

C'est une première étape de test : car ce n'est pas parce que Googlebot peut accéder au contenu que Google va réellement indexer ce contenu, et explorer les liens vers de nouveaux contenus générés en Javascript.

Tester le bon fonctionnement du site en AngularJS

Compte tenu de ses caractéristiques, il s'avère difficile de vérifier si le site fait en AngularJS génère dans le navigateur des internautes un code HTML conforme aux attentes : tout simplement parce que l'affichage du code source fournit peu d'informations.

Pour tester un tel site, il est indispensable de développer (à l'aide d'outils comme Node.js, Phantom.js ou Casper.js) des outils capables de générer une copie (appelée « snapshot ») du code HTML généré par le Javascript, afin de pouvoir vérifier que le contenu attendu est bien là.

C'est par exemple souvent utile pour vérifier que les balises utiles pour le SEO, comme la balise <title> ou la meta description, sont bien générées, et contiennent des infos différentes et adaptées pour chaque page de contenu. Les développeurs de SPA « oublient » bien souvent de générer ces balises, ou ne les génèrent pas correctement, et cela fait donc partie des erreurs souvent rencontrées sur les sites en Angular.

Cette approche demande quand même un bon niveau en Javascript, et une certaine expérience avec les framework Javascript.

Tester l'indexation du contenu

Dans la pratique, on ne pourra pas faire l'économie de tester sur un jeu limité de pages, si Google parvient bien à crawler et indexer tous les contenus. Car même un site bien codé en AngularJS et Javascript, exploitant la méthode PushState, peut connaître des difficultés de référencement dans Google.

Si cela ne fonctionne pas, il s'avère souvent lent et laborieux de trouver l'élément qui « bloque » les analyses de Google. Dans ce cas, le recours au serveur de prérendition s'avère indispensable.

La solution prudente : utiliser un serveur de prérendition

Compte tenu du caractère particulièrement aléatoire et délicat de l'indexation des sites Angular JS, la solution recommandée est d'éviter que Google ait besoin d'exécuter le Javascript pour accéder au contenu. Il est possible d'ajouter une couche logicielle, baptisée serveur de prérendition, qui sera chargée :

- D'exécuter le Javascript du template AngularJS ;
- De générer une copie du HTML produit par Angular ;
- Et c'est cette copie qui sera renvoyée, *a minima* aux moteurs, voire à tous les utilisateurs.

Un site Angular doté d'un serveur de prérendition se met à se comporter comme un site « normal ». Il devient également possible de tester normalement son code source à l'aide d'un crawler de type Xenu ou Screaming Frog.

Il existe différentes solutions de tels moteurs, Open Source (par exemple prerender.io) ou commerciales (comme brombone.com).

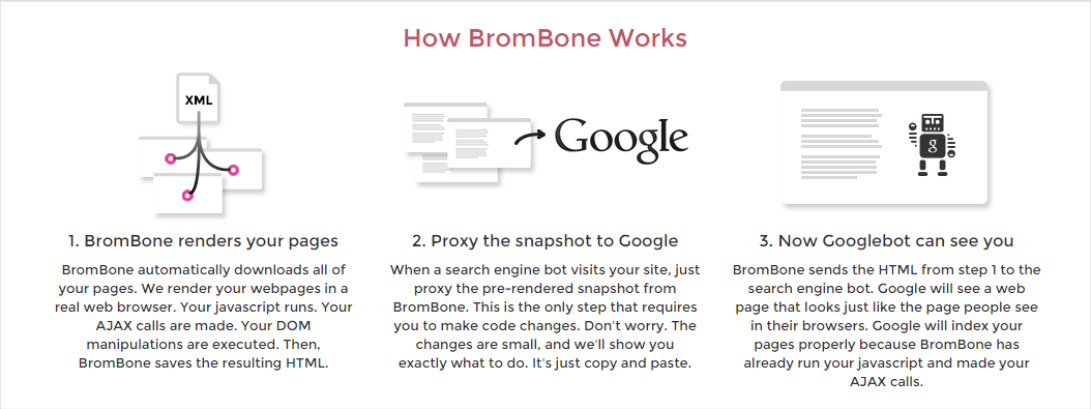

Fig. 6. La présentation du fonctionnement de Brombone.

Peut-on finalement considérer Angular comme une solution « SEO Friendly » ?

En conclusion, si votre site dépend du trafic issu des moteurs de recherche, le recours à ce type de technologie est à déconseiller. C'est d'autant plus vrai si vous n'avez pas sous la main des développeurs expérimentés sur ces technologies, et capables d'implémenter correctement une version compatible SEO du code, un serveur de prérendition, et qui sauront tester et fiabiliser une telle solution.

Qui plus est, seul Google assure aujourd'hui une prise en compte réelle du contenu généré en Javascript. Si vous souhaitez que votre contenu puisse être indexé par Bing, Baidu, Yandex, ou un outil de recherche spécialisé, le recours à des frameworks JS est aujourd'hui formellement déconseillé.

Est-ce que cette situation peut évoluer ? Certainement : la prise en compte du contenu généré en Javascript est récente chez Google, et peut s'améliorer. Les techniques de développement d'application web vont, elles aussi, fortement évoluer. Déjà, des solutions Javascript de type « middleware » commencent à apparaître, qui présentent pour les développeurs les mêmes avantages qu'Angular, mais sans changer le comportement coté *front*.

En attendant, si l'on vous propose de refaire votre site avec ces technologies, soyez avertis : l'impact sur votre référencement peut être funeste. Soyez donc prudents !

Liens utiles

Le site officiel de AngularJS
https://angularjs.org/
Le blog de la communauté AngularJS
http://angularjs.blogspot.fr/
Le site de l'outil Prerender.io
https://prerender.io/

Posts du blog webmasters de Google :

Présentation de la méthode des hash bangs (obsolète) :
http://googlewebmastercentral.blogspot.fr/2009/10/proposal-for-making-ajax-crawlable.html

Annonce des progrès faits par Google dans la prise en compte du Javascript :
http://googlewebmastercentral.blogspot.fr/2014/05/understanding-web-pages-better.html

Google annonce qu'il ne promeut plus la méthode des hashbangs :
http://googlewebmastercentral.blogspot.fr/2015/10/deprecating-our-ajax-crawling-scheme.html

Méthode des hashbangs (obsolète)

Le lien vers l'ancienne spécification, toujours supportée par Google et d'autres moteurs, mais qui est considérée comme obsolète. L'utilisation de cette méthode sur de nouvelles pages/implémentations est déconseillée :
https://developers.google.com/webmasters/ajax-crawling/docs/learn-more
https://developers.google.com/webmasters/ajax-crawling/docs/getting-started
https://developers.google.com/webmasters/ajax-crawling/docs/specification

Méthode employant pushstate() (recommandée)

La recommandation de Bing
https://blogs.bing.com/webmaster/2013/03/21/search-engine-optimization-best-practices-for-ajax-urls/

Et celle de Google : une vidéo de Matt Cutts datant de mars 2013 expliquant les avantages de la méthode pushstate
https://www.youtube.com/watch?v=yiAF9VdvRPw&feature=youtu.be

Philippe YONNET , *Directeur Général de l'agence Search-Foresight, groupe My Media (http://www.search-foresight.com)*