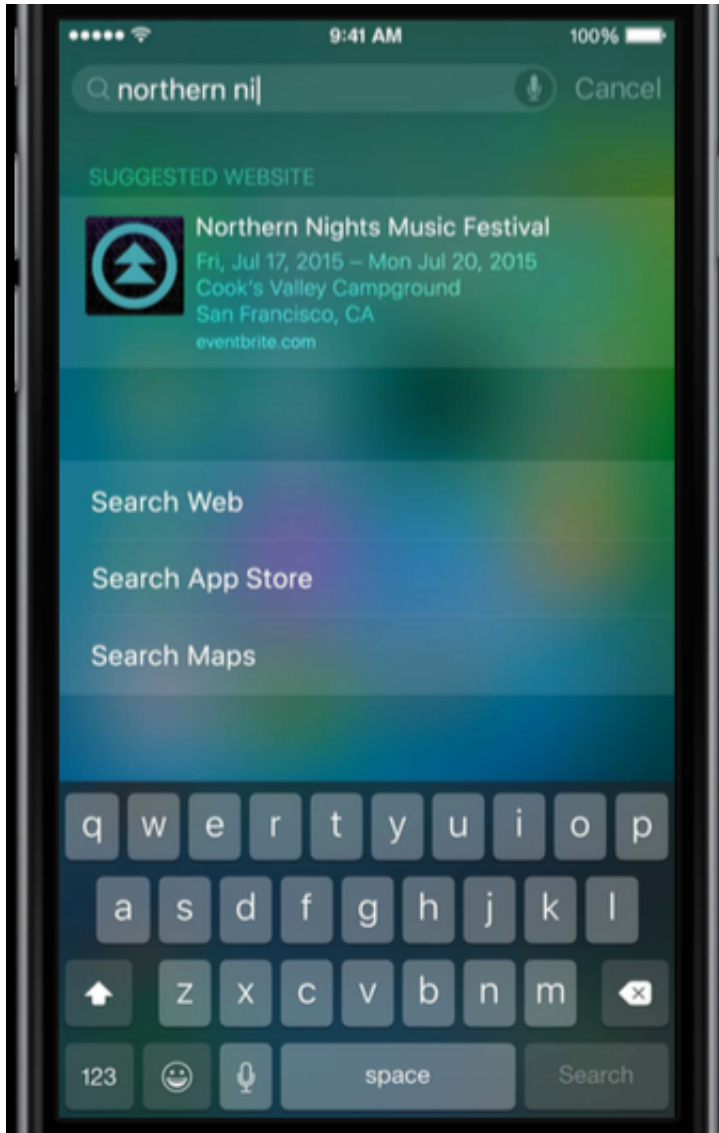
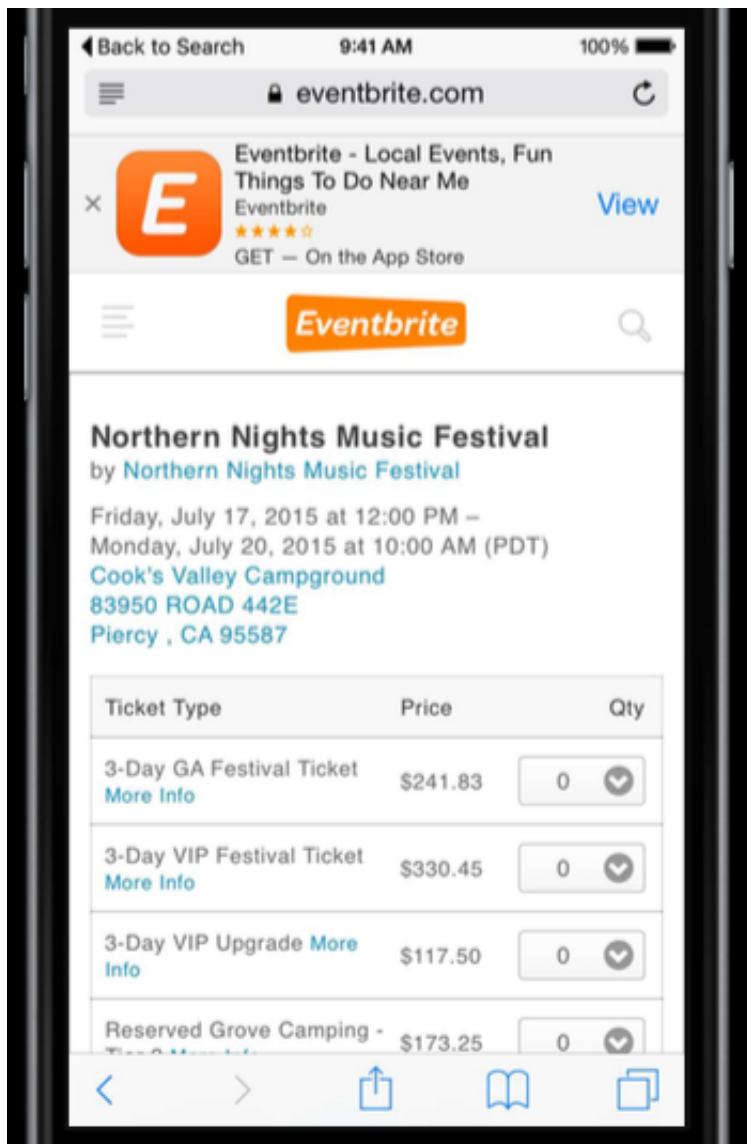


Search APIs – App Deep Linking

在iOS 9 之前Spotlight 搜索功能是对开发者来说是透明的，但是现在开发者可以添加在Spotlight 或者Safari 的搜索结果，甚至在该app 没有安装的情况下也可以。



本地没有安装Eventbrite，可以在搜索结果的网页上显示下载app（不熟悉，请猛戳[Smart App Banner](#)）。



App Search 由 NSUserActivity,CoreSpotlight,Web Markup组成。

App Search

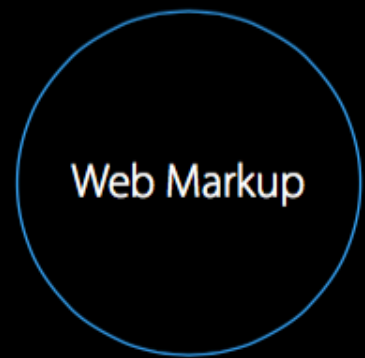
Three APIs



Viewed App Content



Any App Content



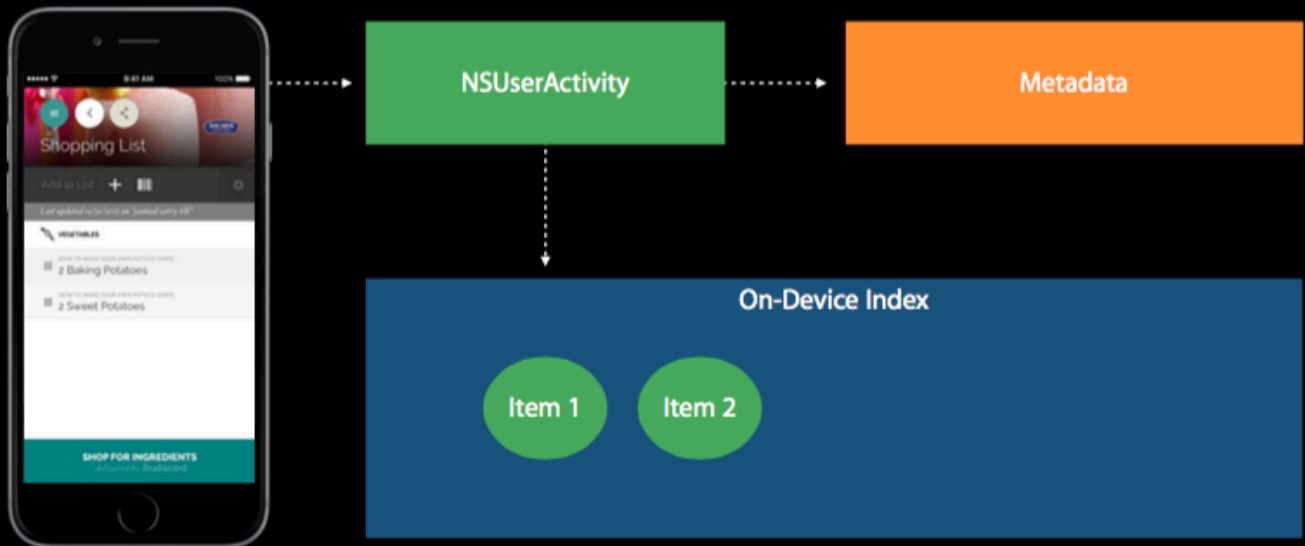
App Content on Web

NSUserActivity

在iOS8 就出现了，主要用户Handoff 功能，现在扩展了它的功能，使它能够像浏览器那样记录你的浏览记录。但是只能用在用户访问过的或者看见过的内容中。一旦某些内容被记录进NSUserActivity，就可以在Spotlight和Safari中同时被搜索到。

NSUserActivity

App search



而且还能通过设置'Eligible For Public Indexing'来让这些被Index的内容传到Apple的云端Cloud Index里，从而实现每个用户都能搜索到这个内容。同时Apple也强调了隐私的保护。并不是所有内容都是Public的，同一个内容需要在云端被Index超过一个限额（具体多少没有公布），才会最后成为Public的内容。所以用户不用担心自己看到的内容成为公众都能搜索的内容。

Create Activity



```
var activity:NSUserActivity = NSUserActivity(activityType:
    "com.yummly.browseRecipe")

activity.title = "Baked Potato Chips"
activity.userInfo = ["id": "http://www.yummly.com/recipe/BPC-983195"]

activity.eligibleForSearch = true

activity.becomeCurrent()
```

Designating Activities Public

```
var activity:NSUserActivity = NSUserActivity(activityType:  
    "com.yummly.browseRecipe")  
  
activity.title = "Baked Potato Chips"  
activity.userInfo = ["id": "http://www.yummly.com/recipe/BPC-983195"]  
activity.eligibleForSearch = true  
activity.eligibleForPublicIndexing = true  
activity.becomeCurrent()
```

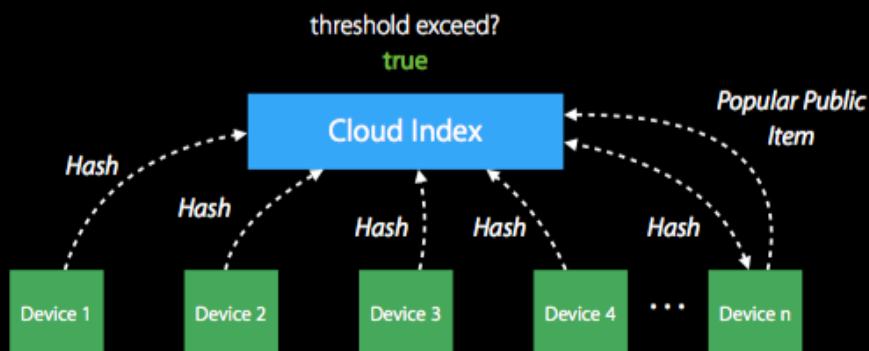
NSUserActivity and Public Indexing

Privacy

Activities are private by default

Designate "public" if searchable activity fields are solely public

Provisions to prevent user-specific activities from being indexed



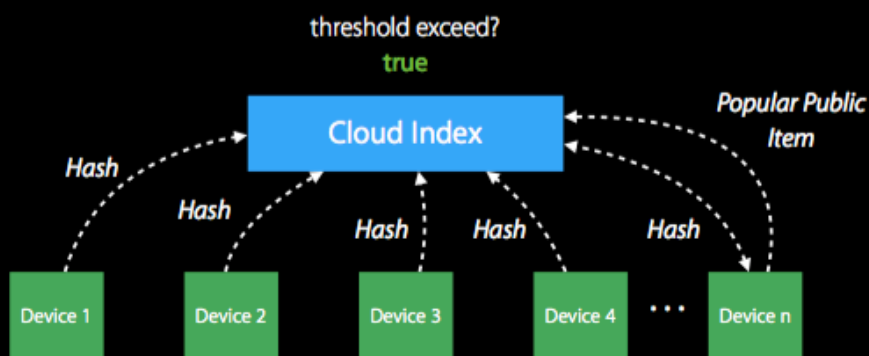
NSUserActivity and Public Indexing

Privacy

Activities are private by default

Designate “public” if searchable activity fields are solely public

Provisions to prevent user-specific activities from being indexed



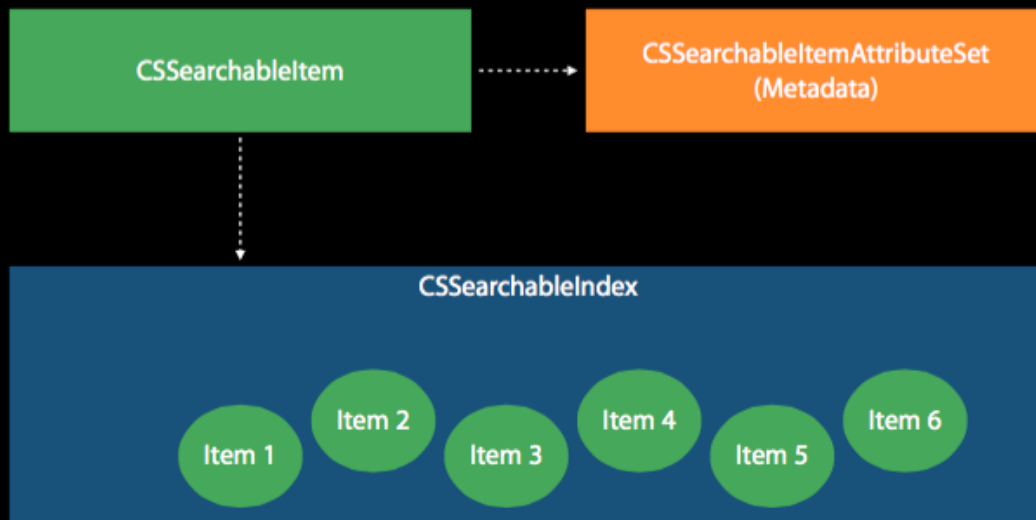
NSUserActivity 让用户更加容易的查看他们以往查看的内容。一方面开发者们也可以充分利用这个功能，更好展现自己的app,让用户体验做到更加好。

CoreSpotlight

就像数据库那样，让开发者可以添加，删除，更新在应用内为用户保存的用于快速搜索的记录。内置的 Mail，Notes等应用也是用这项技术来实现搜索应用内容的功能。一般会是推荐App内用户的文档，图片，信息，等等内容

CoreSpotlight API

NEW



Create Item

```
// Create attributeSet and populate with metadata
let attributeSet = CSSearchableItemAttributeSet(itemContentType: kUTTypeImage as String)
```

```
attributeSet.title = "Haleakala Sunrise"
attributeSet.contentDescription = "May 12, 2015 Maui, Hawaii"
```

```
// Create item with unique identifier; domainIdentifier used to group items
let item = CSSearchableItem(uniqueIdentifier: "1", domainIdentifier: "album-1", attributeSet: attributeSet)
```

```
// Index item
CSSearchableIndex.defaultSearchableIndex().indexSearchableItems([item]) { error in
    if error != nil {
        print(error?.localizedDescription)
    }
    else {
        print("Item indexed!")
    }
}
```

Update

NEW

Same method as adding item to index

```
func indexSearchableItems(items: [CSearchableItem], completionHandler:  
((NSError?) -> Void)?)
```

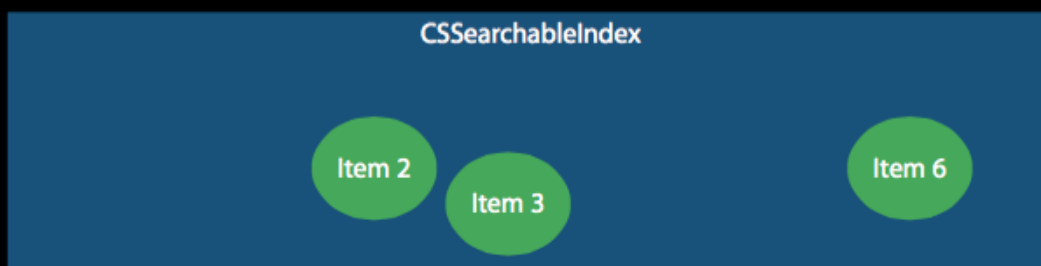


Delete Identifiers

NEW

```
func deleteSearchableItemsWithIdentifiers(identifiers: [String],  
completionHandler: ((NSError?) -> Void)?)
```

```
identifiers = ["1", "4", "5"]
```



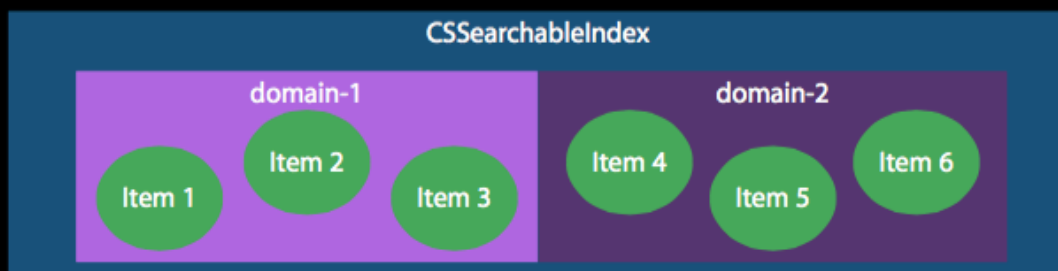
Delete

DomainIdentifier

NEW

```
func deleteSearchableItemsWithDomainIdentifiers(domainIdentifiers: [String],  
completionHandler: ((NSError?) -> Void)?)
```

```
domainIdentifiers = ["domain-2"]
```



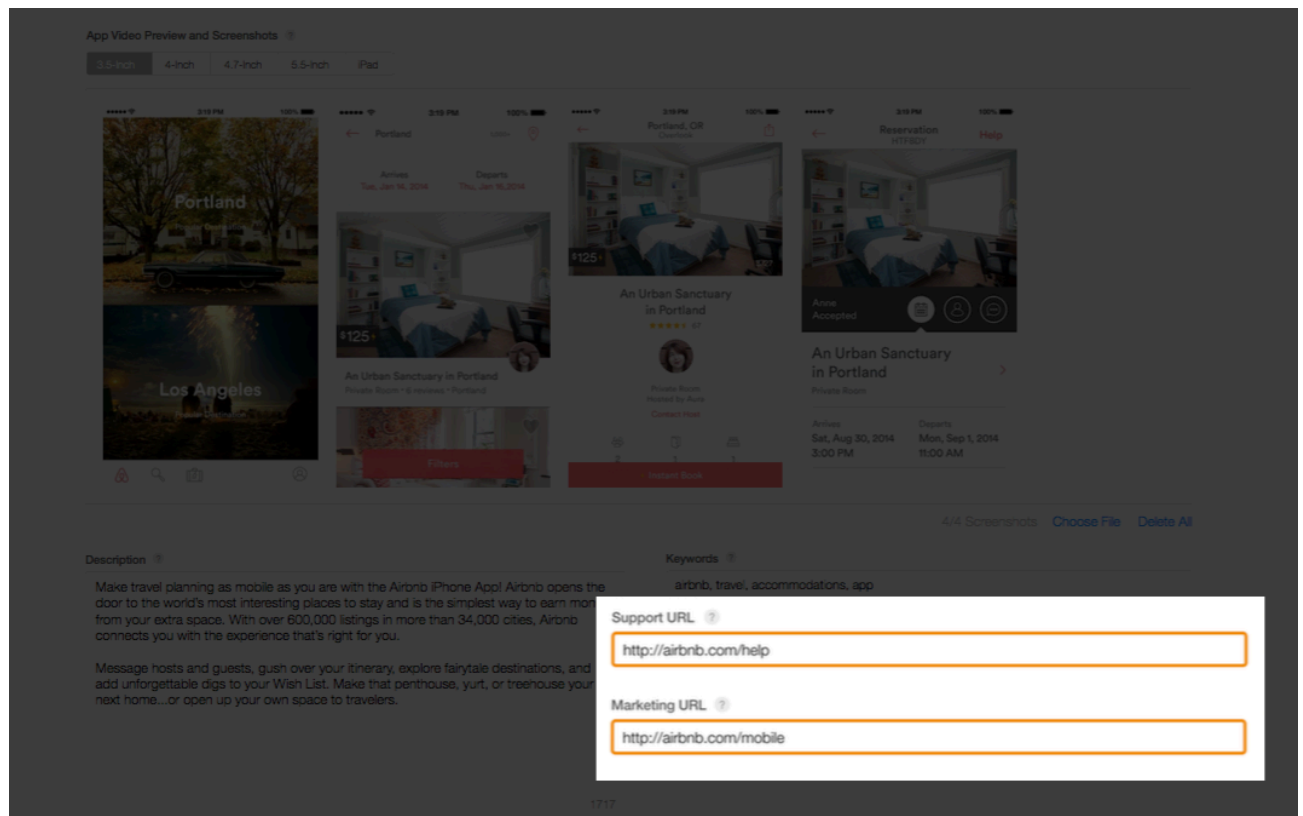
Web Markup

这项功能允许苹果和名为：Applebot search engine 的搜索服务器搜索与app 相关的内容，让用户更加容易的使用搜索功能。支持 "deep linking" and universal link .结合苹果的 Smart App Banners, Twitter Cards 和 Facebook App Links.

要使应用内的网页内容可被搜索，开发者需要设置：

- 1) **Allow Apple to discover and crawl an app's website with the Applebot.**

在iTunes Connect 提交应用的时候，在 Support URL 和 Marketing URL 填写链接。苹果会根据你所提供的URL 来搜索内容。



- 2) Ensure the app's website has the Web markup for deep linking.

Preparing Your Website

Smart App Banners

Smart App Banner provides a clear link into your app
If app is not installed, Safari prompts the user

```
<meta name="myApp" content="app-id=123, app-argument=http://example.com/about">
```

See Promoting Apps with Smart App Banners on the Safari Web Content Guide

Universal Link 是 iOS9 里面的新功能，可以用来取代 smart app banners。但使用 Universal Link 的前提是你的应用已经被安装了。所以建议继续使用 smart app banners。

Preparing Your Website



Universal links

Universal links have several advantages over custom URL schemes

- Unique
- Secure
- Flexible
- Work seamlessly across app and website

Preparing Your Website

Other deep link schemes

Other standards are supported as well, such as Twitter Cards:

```
<meta name="twitter:app:name:iphone" content="myAppName">  
<meta name="twitter:app:id:iphone" content="myAppID">  
<meta name="twitter:app:url:iphone" content="myURL">
```

And Facebook's App Links:

```
<meta property="al:ios:app_name" content="myAppName">  
<meta property="al:ios:app_store_id" content="myAppID">  
<meta property="al:ios:url" content="myURL">
```

- 3) On the other end, make sure the app is set up to handle deep linking.

Preparing Your App

Support deep linking

```
func application(application: UIApplication, openURL url: NSURL,
sourceApplication: String?, annotation: AnyObject) -> Bool {

    // In this example, the URL is http://example.com/profile/?123
    if let components = NSURLComponents(URL: url,
                                        resolvingAgainstBaseURL: true),
        let path = components.path, let query = components.query {
        if path == "/profile" {
            // Pass the profile ID from the URL, to the view controller
            return profileViewController.loadProfile(query)
        }
    }
    return false
}
```

- 4) Add markup for rich data and structured data to be able to surface in search. 搜索结果显示的内容不限于只有title 和 description 了。还可以包含图片，甚至是触发的有两种格式：[Open Graph](#):

Rich Results

Semantic markup calls out key information

Open Graph:

```
<meta property="og:image" content="http://example.com/hello.jpg">
<meta property="og:audio" content="http://example.com/music.m4a">
<meta property="og:video" content="http://example.com/cats.mp4">
```

[Scheme.org](#):

schema.org Microdata:

```
<div itemprop="aggregateRating" itemscope itemType="http://schema.org/
AggregateRating">
    <span itemprop="ratingValue">4</span> stars -
    <span itemprop="reviewCount">250</span> reviews
</div>
```

schema.org JSON-LD:

```
<script type="application/ld+json">
{
  "@context": "http://schema.org",
  "@type": "AggregateRating",
  "ratingValue": "4",
  "reviewCount": "250"
}
</script>
```

Demo

Rich Results

Semantic markup calls out key information

The diagram illustrates how semantic markup is applied to an Airbnb listing card. The listing card contains the following information:

- AIRBNB
- Napa Valley Dreamhouse in Napa
- Entire home/apt. 3 bedroom, 2 bath
- \$650 per night
- ★★★★★ 29 reviews

The corresponding HTML markup is shown below:

- `<title>Napa Valley Dream House in Napa</title>` points to the listing title.
- `<meta itemprop="priceCurrency" content="USD">` and `<meta itemprop="price" content="650">` point to the price.
- `29` points to the review count.
- `<meta itemProp="ratingValue" content="5.0">` points to the star rating.
- `<meta property="og:image" content="http://...">` points to the listing image.

Supported schemas

- AggregateRating
- Offers
- Price Range
- InteractionCount
- Organization
- Recipe
- SearchAction
- ImageObject

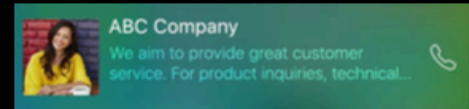
Actions Current supported actions are:

- Dialing a phone number
- Getting directions to an address
- Playing audio or video

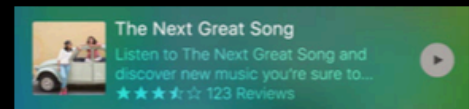
Rich Results

Actions

```
<div itemscope itemtype=
  "http://schema.org/Organization">
  <span itemprop="telephone">
    (408) 123-4567</span>
</div>
```



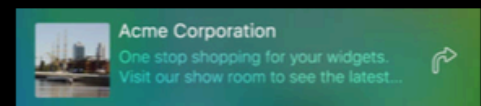
```
<div itemscope itemtype=
  "http://schema.org/AudioObject">
  <meta itemprop="contentUrl"
    content="http://abc.com/x.m4a" />
</div>
```



Rich Results

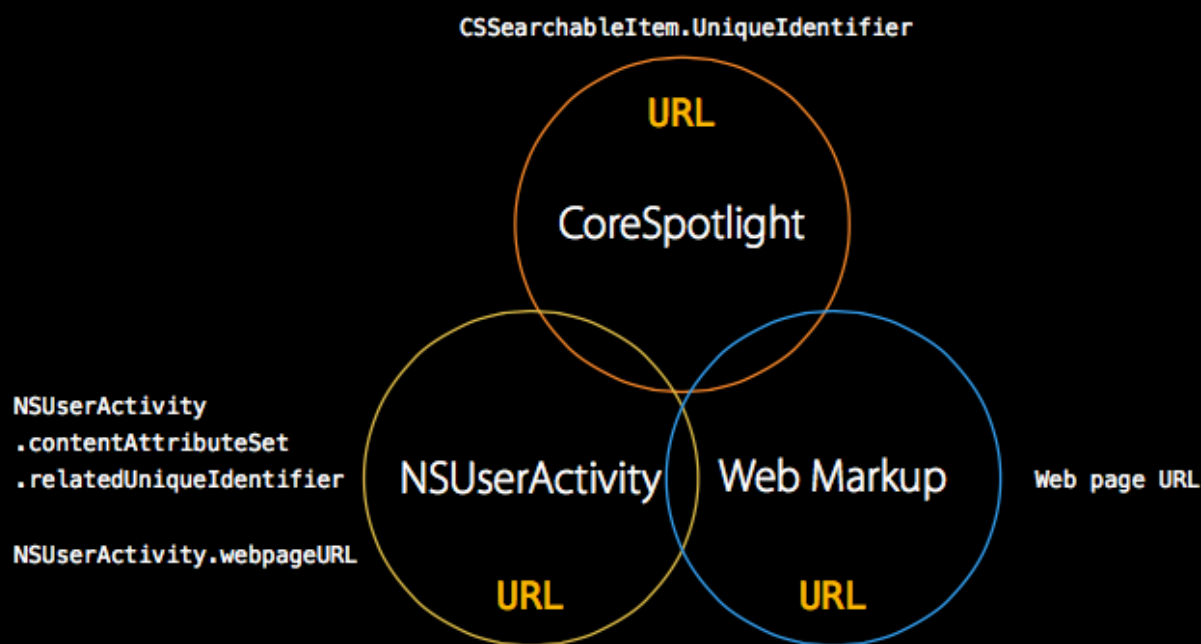
Actions

```
<div itemprop="address" itemscope
  itemtype="http://schema.org/
  PostalAddress">
  <span itemprop="streetAddress">
    200 Innovation Way
  </span>
  <span itemprop="addressLocality">
    San Francisco</span>
  <span itemprop="addressRegion">
    CA</span>
  <span itemprop="postalCode">
    94108</span>
</div>
```



CoreSpotlight 让你的应用内的内容可以让用户更加容易搜索到。例如微信，可以把所有的聊天信息都加上 index，这样用户就可以在spotlight 里面搜索内容。**Web markup** ,可以让用户没有安装你的应用的前提下，搜索到你的应用的内容，因此提高app 的下载率。**NSUserActivity 'Eligible For Public Indexing's** 使被Index 的内容传到Apple的云端Cloud Index里，可以让更多的用户搜索到Public Indexing's 的内容。三者其实可以协同工作的，同一个app，同一个内容被三者同时使用，可以通过设置统一的ID(URL) 来实现。

Relevance: Linking APIs



搜索内容的排名对于用户和开发者来说是透明的，提高搜索的排名。可以通过以下几点。

- 1) URL Popularity. (URL 在应用内被点击或者出现的次数)
- 2) Activities (应用内的相对用的content出现的频率，可以用过使用NSUserActivity来提高)
- 3) Engagements.(用户与搜索的内容之间的交互，例如点击)

Optimizing Relevance

Best practices

Create great apps and content

Adopt API provisions that contribute to relevance

- NSUserActivity
- NSUserActivity public indexing where appropriate
- Schema markup for ratings and reviews

Follow User Interface Guidelines

