iTunes Store
Web Service Search API

version 2.1

06/27/2007

iTunes Store
Web Service Search API

Overview

The iTunes Store Web Service Search API allows you to place search fields in your website to search for content within the iTunes Store. You can search for a variety of iTunes Store content; including movies, podcasts, music, music videos, audiobooks, and TV shows. You can also call an ID-based lookup request to create mappings between your content library and the iTunes Store content library.

The iTunes Affiliate Program allows you to earn a commission on qualifying revenue generated by clicks initiated from your website into the iTunes Store. For more information on the Affiliate Program, see http://www.apple.com/itunes/affiliates.

Searching the iTunes Store

To search for iTunes Store content from a field in your website and display the results in your website, you must create a search field that passes a fully-qualified URL content request to the iTunes Store, parse the JavaScript Object Notation (JSON) format returned from the search, and display the results in your website.

The fully-qualified URL must have the following format:

```
http://ax.phobos.apple.com.edgesuite.net/WebObjects/MZStoreServices.woa/wa/
wsSearch?parameterkeyvalue
```

Where *parameterkeyvalue* can be one or more parameter key and value pairs indicating the details of your query.

To construct a parameter key and value pair, you must concatenate each parameter key with an equal sign (=) and a value string. For example: `key1=value1`. To create a string of parameter key and value pairs, you must concatenate each pair using an ampersand (&). For example:

```
key1=value1&key2=value2&key3=value3
```

Note: When creating search fields and scripts for your website, you should use dynamic script tags for your xmlhttp script call requests. For example:

```
<script src=”http://...wsSearch?parameterkeyvalue&callback=”{name of JavaScript
function in webpage}”/>
```

The following table defines the parameter keys and values you can specify to search for content within the iTunes Store:

<table>
<tr><th>Parameter Key</th><th>Description</th><th>Required</th><th>Values</th></tr>
<tr><td>term</td><td>The URL-encoded text string you want to search for in the iTunes Store. For example: jack+johnson.</td><td>Y</td><td>Any URL-encoded text string.
<b>Note</b>: URL encoding replaces spaces with the plus (+) character and all characters except the following are encoded: letters, numbers, periods (.), dashes (-), underscores (_), and asterisks (*).</td></tr>
<tr><td>country</td><td>The two-letter country code for the iTunes Store you want to search. The search uses the default store front for the specified country. For example: US.
The default is US.</td><td>Y</td><td>See http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2 for a list of ISO Country Codes.</td></tr>
<tr><td>media</td><td>The media type you want to search for in the iTunes Store. For example: movie.
The default is all.</td><td>N</td><td>movie, podcast, music, musicVideo, audiobook, shortFilm,tvShow, all</td></tr>
<tr><td>entity</td><td>The type of results you want the iTunes Store to return, relative to the specified media type. For example: movieArtist for a movie media type search.
The default is the track entity associated with the specified media type.</td><td>N</td><td>The following entities are available for each media type:
<table>
<tr><td>movie</td><td>movieArtist, movie</td></tr>
<tr><td>podcast</td><td>podcastAuthor, podcast</td></tr>
<tr><td>music</td><td>musicArtist, musicTrack, album, musicVideo, mix</td></tr>
<tr><td>musicVideo</td><td>musicArtist, musicVideo</td></tr>
<tr><td>audiobook</td><td>audiobookAuthor, audiobook</td></tr>
<tr><td>shortFilm</td><td>shortFilmArtist, shortFilm</td></tr>
<tr><td>tvShow</td><td>tvEpisode, tvSeason</td></tr>
<tr><td>all</td><td>movie, album, allArtist, podcast, musicVideo, mix, audiobook, tvSeason, allTrack</td></tr>
</table>
</td></tr>
</table>

Parameter Key	Description	Required	Values	
attribute	The attribute you want to search for in the iTunes Store, relative to the specified media type. For example, if you want to search for an artist by name specify entity=allArtist&attribute=allArtistTerm. In this example, if you search for term=maroon, iTunes returns "Maroon 5" in the search results, instead of all artists who have ever recorded a song with the word "maroon" in the title. The default is all attributes associated with the specified media type.	N	The following attributes are available for each media type:	
			movie	actorTerm, genreIndex, artistTerm, shortFilmTerm, producerTerm, ratingTerm, directorTerm, releaseYearTerm, featureFilmTerm, movieArtistTerm, movieTerm, ratingIndex, descriptionTerm
			podcast	titleTerm, languageTerm, authorTerm, genreIndex, artistTerm, ratingIndex, keywordsTerm, descriptionTerm
			music	mixTerm, genreIndex, artistTerm, composerTerm, albumTerm, ratingIndex, songTerm, musicTrackTerm
			musicVideo	genreIndex, artistTerm, albumTerm, ratingIndex, songTerm
			audiobook	titleTerm, authorTerm, genreIndex, ratingIndex
			shortFilm	genreIndex, artistTerm, shortFilmTerm, ratingIndex, descriptionTerm
			tvShow	genreIndex, tvEpisodeTerm, showTerm, tvSeasonTerm, ratingIndex, descriptionTerm
			all	actorTerm, languageTerm, allArtistTerm, tvEpisodeTerm, shortFilmTerm, directorTerm, releaseYearTerm, titleTerm, featureFilmTerm, ratingIndex, keywordsTerm, descriptionTerm, authorTerm, genreIndex, mixTerm, allTrackTerm, artistTerm, composerTerm, tvSeasonTerm, producerTerm, ratingTerm, songTerm, movieArtistTerm, showTerm, movieTerm, albumTerm
callback	The name of the Javascript callback function you want to use when returning search results to your website. For example: wsSearchCB.	Y, for cross-site searches	wsSearchCB	
limit	The number of search results you want the iTunes Store to return. For example: 25. The default is 100.	N	1 to 200	
lang	The language, English or Japanese, you want the iTunes Store to use when returning search results. Specify the language using the five-letter codename. For example: en_us. The default is en_us (English).	N	en_us, ja_jp	

Parameter Key	Description	Required	Values
version	The iTunes Store search result key version you want to receive back from your search. The default is 2.	N	1, 2
explicit	A flag indicating whether or not you want the iTunes Store to include explicit content in your search results. The default is Yes.	N	Yes, No

Notes:

- It is critical to generate your URLs correctly in order to get the search reference recognized as Affiliate Program revenue in the iTunes Store
- To improve response times, minimize the number of search results the iTunes Store returns by specifying an appropriate value for the limit parameter key.
- Large websites should set up caching logic for the search and lookup requests sent to the iTunes Store. For an illustration, see Caching Architecture later in this document.

Search Examples

The following are examples of fully-qualified URLs for specific iTunes Store search requests:

- To search for all Jack Johnson audio and video content (movies, podcasts, music, music videos, audiobooks, short films, and tv shows), your URL would look like the following: http://ax.phobos.apple.com.edgesuite.net/WebObjects/MZStoreServices.woa/wa/wsSearch?term=jack+johnson
- To search for all Jack Johnson audio and video content and return only the first 25 items, your URL would look like the following: http://ax.phobos.apple.com.edgesuite.net/WebObjects/MZStoreServices.woa/wa/wsSearch?term=jack+johnson&limit=25
- To search for only Jack Johnson music videos, your URL would look like the following: http://ax.phobos.apple.com.edgesuite.net/WebObjects/MZStoreServices.woa/wa/wsSearch?term=jack+johnson&entity=musicVideo
- To search for all Jim Jones audio and video content and return only the results from the Canada iTunes Store, your URL would look like the following: http://ax.phobos.apple.com.edgesuite.net/WebObjects/MZStoreServices.woa/wa/wsSearch?term=jim+jones&country=ca

Lookup Examples

You can also create a *lookup* request to search for content in the iTunes Store based on iTunes IDs and All Music Guide (AMG) IDs, as opposed to search terms. ID-based lookups are faster and contain fewer false-positive results.

The following are examples of fully-qualified URLs for specific iTunes Store lookup requests:

- To look up Jack Johnson by iTunes artist ID, your URL would look like the following: http://ax.phobos.apple.com.edgesuite.net/WebObjects/MZStoreServices.woa/wa/wsLookup?id=909253
- To look up Jack Johnson by AMG artist ID, your URL would look like the following: http://ax.phobos.apple.com.edgesuite.net/WebObjects/MZStoreServices.woa/wa/wsLookup?amgArtistId=468749
- To look up multiple artists by their AMG artist IDs, your URL would look like the following: http://ax.phobos.apple.com.edgesuite.net/WebObjects/MZStoreServices.woa/wa/wsLookup?amgArtistId=468749,5723
- To look up all albums for Jack Johnson, your URL would look like the following: http://ax.phobos.apple.com.edgesuite.net/WebObjects/MZStoreServices.woa/wa/wsLookup?id=909253&entity=album
- To look up multiple artists by their AMG artist IDs and get each artist's top 5 songs, your URL would look like this: http://ax.phobos.apple.com.edgesuite.net/WebObjects/MZStoreServices.woa/wa/wsLookup?amgArtistId=468749,5723&entity=album&limit=5

Understanding Search Results

The iTunes Store returns your search results in JavaScript Object Notation (JSON) format. JSON is built on two structures:

- A collection of name/value pairs, also known as an *object*; a concept similar to a Java Map object, a Javascript Dictionary, or a Pearl/Ruby hash. An object is an unordered set of name/value pairs, beginning with a left brace ({) and ending with a right brace (}). Each name is following by a colon (:) and the name/value pairs are separated by commas (,).
- An ordered list of values, also known as an *array*. An array is an ordered collection of values, beginning with a left bracket ([ ) and ending with a right bracket ( ]). Values are separated by commas (,).

All JSON results are returned in UTF-8 encoded format. You must parse the JSON results and use the information in your search fields, or scripts, to display the search results properly in your website. For more information on JSON and the details of values, strings, numbers, and more, see http://www.json.org.

The following example displays the JSON results for a song in the iTunes Store:

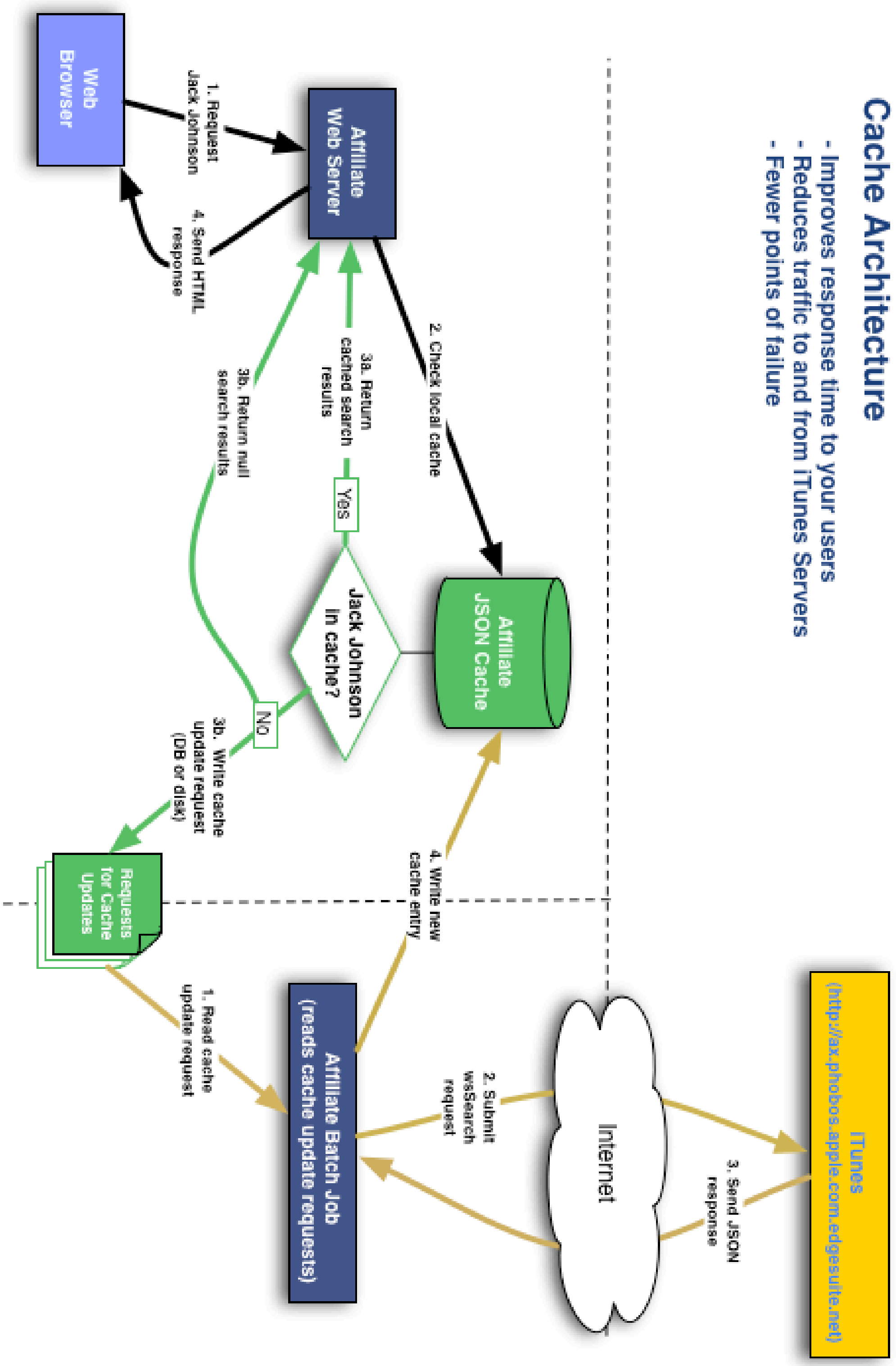
```
{"wrapperType":"track",
 kind":"song",
 artistId":909253,
 collectionId":120954021,
 trackId":120954025,
 artistName":"Jack Johnson",
 collectionName":"Sing-a-Longs and Lullabies for the Film Curious George",
 trackName":"Upside Down",
 collectionCensoredName":"Sing-a-Longs and Lullabies for the Film Curious George",
 trackCensoredName":"Upside Down",
 artistViewUrl":"http://phobos.apple.com/WebObjects/MZStore.woa/wa/viewArtist?
id=909253",
 collectionViewUrl":"http://phobos.apple.com/WebObjects/MZStore.woa/wa/viewAlbum?
i=120954025&id=120954021&s=143441",
 trackViewUrl":"http://phobos.apple.com/WebObjects/MZStore.woa/wa/viewAlbum?
i=120954025&id=120954021&s=143441",
 previewUrl":"http://a1099.phobos.apple.com/r10/Music/f9/54/43/
mzi.gqvqlvcq.aac.p.m4p",
 artworkUrl60":"http://a1.phobos.apple.com/r10/Music/3b/6a/33/mzi.qzdqwsel.
60x60-50.jpg",
 artworkUrl100":"http://a1.phobos.apple.com/r10/Music/3b/6a/33/mzi.qzdqwsel.
100x100-75.jpg",
 collectionPrice":10.99,
 trackPrice":0.99,
 collectionExplicitness":"notExplicit",
 trackExplicitness":"notExplicit",
 discCount":1,
 discNumber":1,
 trackCount":14,
 trackNumber":1,
 trackTimeMillis":210743,
 country":"USA",
 currency":"USD",
 primaryGenreName":"Rock"}
```

The following table defines the JSON result keys and values:

Result Key	Description	Returned	Return Values and Examples
wrapperType	The name of the object returned by the search request.	Y	track, collection, artist For example: track.
*explicitness	The Recording Industry Association of America (RIAA) parental advisory for the content returned by the search request. For more information, see http://phobos.apple.com/WebObjects/MZStore.woa/wa/parentalAdvisory.	Y	explicit (explicit lyrics, possibly explicit album cover), cleaned (explicit lyrics “bleeped out”), notExplicit (no explicit lyrics) For example: “trackExplicitness”:”notExplicit”.
kind	The kind of content returned by the search request.	Y	book, album, coached-audio, feature-movie, interactive-booklet, music-video, pdf podcast, podcast-episode, software-package, song, tv-episode, artist For example: song.

Result Key	Description	Returned	Return Values and Examples
trackName	The name of the track, song, video, TV episode, and so on returned by the search request.	Y	For example: "Banana Pancakes".
artistName	The name of the artist returned by the search request.	Y	For example: Jack Johnson.
collectionName	The name of the album, TV season, audiobook, and so on returned by the search request.	Y	For example: "In Between Dreams".
*censoredName	The name of the album, TV season, audiobook, and so on returned by the search request, with objectionable words *'d out. **Note**: Artist names are never censored.	Y	For example: "S**t Happens".
artworkUrl100 artworkUrl60	A URL for the artwork associated with the returned media type, sized to 100x100 pixels or 60x60 pixels.	Only returned when artwork is available	For example: "http://a1.phobos.apple.com/jp/r10/Music/y2005/m06/d03/h05/s05.oazjtxkw.100x100-75.jpg".
*viewURL	A URL for the content associated with the returned media type. You can click the URL to view the content in the iTunes Store.	Y	For example: "http://phobos.apple.com/WebObjects/MZStore.woa/wa/viewAlbum?i=68615807&id=68615813&s=143462".
previewUrl	A URL referencing the 30-second preview file for the content associated with the returned media type. The iTunes Store encodes the files for playback only by iTunes, QuickTime, or iTunes and QuickTime plugins.	Only returned when media type is track	For example: "http://a392.phobos.apple.com/jp/r10/Music/y2005/m06/d03/h05/s05.zdzqlufu.p.m4p".
trackTimeMillis	The returned track's time in milliseconds.	Only returned when media type is track	

Caching Architecture


Cache Architecture
- Improves response time to your users
- Reduces traffic to and from iTunes Servers
- Fewer points of failure
Web Browser
1. Request Jack Johnson
4. Send HTML response
Affiliate Web Server
2. Check local cache
3a. Return cached search results
Yes
3b. Return null search results
Affiliate JSON Cache
Jack Johnson in cache?
No
3b. Write cache update request (DB or disk)
Requests for Cache Updates
4. Write new cache entry
1. Read cache update request
Affiliate Batch Job (reads cache update requests)
2. Submit wsSearch request
Internet
3. Send JSON response
iTunes (http://ax.phobos.apple.com.edgesuite.net)