



Mobile App Solution for ads.txt

Draft for public comment | June 2018

Please email openmedia@iabtechlab.com with feedback by July 6, 2018

Mobile App Solution for ads.txt and ads.cert

This document details the proposed mobile app guidance for ads.txt and other future web-based resources that app developers may want to use to fight fraud or work towards transparency in the supply chain. This document will be out for a 30-day public comment period, in which the IAB Technology Laboratory's OpenRTB working group is asking for public review to determine if there is buy-in for the proposed solution, particularly from the App Stores, and also asking for technical feedback to refine and improve the solution.

Ads.txt, a simple, flexible and secure method that publishers and distributors can use to publicly declare the companies they authorize to sell their digital inventory, has seen over 2 million domains of publishers adopt the spec since its marketplace release in June 2017. The ads.txt specification is stewarded by the OpenRTB working group at IAB Tech Lab.

Expected audience for this document: App publishers and ad tech vendors interested in enabling supply-chain security via ads.txt and the upcoming ads.cert ([See OpenRTB for ads.cert](#)) for in-app bid requests. With this release, the working group is also expecting app stores to review and provide feedback.

Public Comment

The 30-day public comment period will end on July 6, 2018. Comments from the public can be sent to openmedia@iabtechlab.com. Also, IAB Tech Lab members are welcome to join the discussion in the OpenRTB working group.

After consideration of comments from the public, and after resolving the open issues below, IAB Tech Lab will release a final document for mobile app adoption of ads.txt. There is not currently a set publication date for the final document.

Open Issues

1. Determining a global name space for mobile apps: When buyers receive an ad request today from a mobile app they do not know for sure which app store it came from (e.g. for Android apps), thus buyers don't know for sure which app it came from. To enable accurate lookup of ads.txt information, a "global name space" would identify which app store to fetch the relevant ads.txt information from. Note: This is important for other additional cases beyond ads.txt, for example, brand safety enforcement).
2. App stores should confirm the most effective implementation, supporting the digital ad industry's use of app metadata for the purpose of better protection against invalid traffic and ad fraud with ads.txt.
3. Selection of which solution to go forward with (See Solution A, Solution B, Solution C on page 6 of this document).

About IAB Tech Lab

The IAB Technology Laboratory (Tech Lab) is a non-profit research and development consortium that produces and provides standards, software, and services to drive growth of an effective and sustainable global digital media ecosystem. Comprised of digital publishers and ad technology firms, as well as marketers, agencies, and other companies with interests in the interactive marketing arena, IAB Tech Lab aims to enable brand and media growth via a transparent, safe, effective supply chain, simpler and more consistent measurement, and better advertising experiences for consumers, with a focus on mobile and TV/digital video channel enablement. The IAB Tech Lab portfolio includes the DigiTrust real-time standardized identity service designed to improve the digital experience for consumers, publishers, advertisers, and third-party platforms. Board members include AppNexus, ExtremeReach, Google, GroupM, Hearst Digital Media, Integral Ad Science, Index Exchange, LinkedIn, MediaMath, Microsoft, Moat, Pandora, PubMatic, Quantcast, Telaria, The Trade Desk, and Yahoo! Japan. Established in 2014, the IAB Tech Lab is headquartered in New York City with an office in San Francisco and representation in Seattle and London.

Learn more about IAB Tech Lab here: <https://www.iabtechlab.com/>

License

The IAB Tech Lab is licensed under a Creative Commons Attribution 3.0 License. To view a copy of this license, visit creativecommons.org/licenses/by/3.0/ or write to Creative Commons, 171 Second Street, Suite 300, San Francisco, CA 94105, USA.



Table of Contents:

Mobile App Solution for ads.txt and ads.cert	2
Public Comment	2
Open Issues	2
About IAB Tech Lab	3
License	3
Table of Contents:	4
Problem	5
Solution Concept	5
Current Assets	5
How to get the files once a domain is identified	5
Example Process	6
Solution Proposal A: Designated Field from App Store	7
Solution Proposal B: App Store API for Retrieving Data	7
Solution Proposal C: Independent Service	7
Appendix A: Current API Solutions	8
Apple API	8
Appendix B: Proposed AppStore API	12

Problem

With the launch of [ads.txt](#) and the upcoming launch of [publisher signatures](#) (ads.cert within OpenRTB 3.0), there is a need for someone receiving a bid request to be able to obtain files from the publisher (ads.txt and ads.cert files). For web supply, this is relatively trivial as the domain field is in the bid request and files can be found at domain.com/ads.txt and domain.com/ads.cert. However, for in-app requests the domain field may not be present. Instead, the primary identifier for the app is the bundle id or application package. So, the question is how can we obtain these files, given a bundle id or package name? Although the most obvious solution would be to add a domain to the bid request, this provides *no actual security* as any fraudulent intermediary can change that domain to point to their own domain with their own ads.txt and ads.cert files.

Solution Concept

Both large app stores (Apple App Store and Google Play) and many of the smaller ones, allow developers to fill in a domain when they add their app to the store. This domain could be used to obtain the ads.txt and ads.cert files. The buy-side would query the app store to get the domain value, then look up the ads.txt and ads.cert there.

Current Assets

Although some of the app stores currently offer some form of URL to get metadata about the app (see Appendix A), this is not consistently supported nor designed for this use case. The IAB will request all large app stores to implement a standard field or designate a standard field (see below) for this purpose.

- For Apple's iTunes API, the 'developerWebsite' or 'sellerUrl' field would be an ideal existing place to translate an App's store ID to the canonical website address. These fields are not well populated, however.
- For the Google Play store, data on the app page contains a developer website link, though this needs standardizing to be reliable as well as an accessible API.
- Amazon's Appstore has no externally accessible webpage of apps, and the app publisher API is not fit for this purpose as it manages only a single publisher's apps.

How to get the files once a domain is identified

In theory, you could just go to the site given and get /ads.txt or /ads.cert. However, frequently this URL refers to the company rather than a specific website for the individual app, and that company may offer multiple apps. Thus, the suggested solution is to get the file at:

- Apple:
 - \$domain/\$bundleid/\$filename
- Android:
 - \$domain/\$applicationpackage/\$filename

where

- \$domain is the domain retrieved via the above with any path after the domain removed
- \$bundleid is the iOS bundleid seen in the bid request - NOTE that while some exchanges operate on the App Store ID, this is meant as the bundleid. Should a bidder receive an App Store ID, they will need to convert it to a bundleid first. Note that the case received in the bid request should be maintained as this field may be case-sensitive.
- \$applicationpackage is the application package name seen in the bid request. Note that the case received in the bid request should be maintained as this field may be case-sensitive.
- \$filename is ads.txt or ads.cert.\$v (where \$v is the certificate version, as sent in the bid request)

Note: developers with many apps will likely implement this as a service, rather than a manually-curated set of files and directories. It is possible third-parties would develop such a service.

Example Process

An example of the process would be as follows:

Step 1: Request official domain of a given app's ID. Here we are using the Fortnite App (id=1261357853).

```
http://itunes.apple.com/lookup?id=1261357853
```

Step 2: Extract the sellerURL from the API response:

```
"sellerUrl":"http://www.fortnite.com"
```

Step 3: Fetch the ads.txt or ads.cert file

```
http://www.fortnite.com/id1261357853/ads.txt
```

```
# Ads.txt file for Fortnite App  
greenadexchange.com, 12345, DIRECT, d75815a79  
silverssp.com, 9675, RESELLER, f496211  
blueadexchange.com, XF436, DIRECT  
orangeexchange.com, 45678, RESELLER
```

Solution Proposal A: Designated Field from App Store

The first proposed solution is to have an officially endorsed and supported field within the description of the app in the store's API and web pages that explicitly identifies the website domain of the app.

With this solution, [the IAB Tech Lab reference crawler](#) could access the API or spider the app store's web page of the app, identifying the official domain. It would then construct the request to the domain to access that app's ads.txt file.

Solution Proposal B: App Store API for Retrieving Data

The second proposed solution is to request that all large app stores implement a standardized API for retrieving an app's ownership meta-data when given one or more bundleids or package names.

Solution Proposal C: Independent Service

The third proposed solution is to put responsibility into a separate entity to provide a [DNS](#)-like service that could map app store IDs to website domains, implementing the proposed central API. The entity would require that the app publisher make some demonstrable change to their app store page as well as an asset on the website domain to prove and verify common control.

Appendix A - Current API Solutions

Apple offers an [official API](#) to obtain the domain if given a bundle id. A [third-party company](#) also offers a version of this API.

Android does not currently offer such an API, officially. There is one [unofficial one](#) and the [same third-party company](#) offers it.

Apple API

Fetching by iTunes store ID, request `http://itunes.apple.com/lookup?id=$theid`:

Fetching by bundle/package, request

`http://itunes.apple.com/lookup?bundleId=$packagebundle`:

```
{
  "resultCount": 1,
  "results": [{
    "artistViewUrl":
"https://itunes.apple.com/us/developer/zynga-
inc/id295913422?mt=8&uo=4",
    "artworkUrl60":
"http://is4.mzstatic.com/image/thumb/Purple128/v4/47/cb/ed/47cbcdc2-
a148-6940-a891-0b2d03e4e172/source/60x60bb.jpg",
    "artworkUrl100":
"http://is4.mzstatic.com/image/thumb/Purple128/v4/47/cb/ed/47cbcdc2-
a148-6940-a891-0b2d03e4e172/source/100x100bb.jpg",
    "ipadScreenshotUrls":
["http://is4.mzstatic.com/image/thumb/Purple118/v4/77/be/1b/77be1b6d-
389e-57df-f484-ae2a19438b53/source/576x768bb.jpg",
"http://is1.mzstatic.com/image/thumb/Purple118/v4/cd/06/ba/cd06ba6c-
23f0-ffda-d60a-fdbbed758631/source/576x768bb.jpg",
"http://is4.mzstatic.com/image/thumb/Purple128/v4/5b/90/ef/5b90ef38-
0996-cab8-90f2-d3851ca4eb65/source/576x768bb.jpg",
"http://is1.mzstatic.com/image/thumb/Purple128/v4/e5/5f/e9/e55fe986-
3c3a-ee2f-ce08-c8b28bfe5392/source/576x768bb.jpg",
"http://is2.mzstatic.com/image/thumb/Purple118/v4/10/3f/c4/103fc430-
cc7d-8748-629d-add0c0de2b76/source/576x768bb.jpg"],
    "appletvScreenshotUrls": [],
    "artworkUrl512":
"http://is4.mzstatic.com/image/thumb/Purple128/v4/47/cb/ed/47cbcdc2-
a148-6940-a891-0b2d03e4e172/source/512x512bb.jpg",
    "isGameCenterEnabled": false,
    "kind": "software",
    "features": ["iosUniversal"],
```

```

    "supportedDevices": ["iPad2Wifi-iPad2Wifi", "iPad23G-
iPad23G", "iPhone4S-iPhone4S", "iPadThirdGen-iPadThirdGen",
"iPadThirdGen4G-iPadThirdGen4G", "iPhone5-iPhone5",
"iPodTouchFifthGen-iPodTouchFifthGen", "iPadFourthGen-iPadFourthGen",
"iPadFourthGen4G-iPadFourthGen4G", "iPadMini-iPadMini", "iPadMini4G-
iPadMini4G", "iPhone5c-iPhone5c", "iPhone5s-iPhone5s", "iPadAir-
iPadAir", "iPadAirCellular-iPadAirCellular", "iPadMiniRetina-
iPadMiniRetina", "iPadMiniRetinaCellular-iPadMiniRetinaCellular",
"iPhone6-iPhone6", "iPhone6Plus-iPhone6Plus", "iPadAir2-iPadAir2",
"iPadAir2Cellular-iPadAir2Cellular", "iPadMini3-iPadMini3",
"iPadMini3Cellular-iPadMini3Cellular", "iPodTouchSixthGen-
iPodTouchSixthGen", "iPhone6s-iPhone6s", "iPhone6sPlus-iPhone6sPlus",
"iPadMini4-iPadMini4", "iPadMini4Cellular-iPadMini4Cellular",
"iPadPro-iPadPro", "iPadProCellular-iPadProCellular", "iPadPro97-
iPadPro97", "iPadPro97Cellular-iPadPro97Cellular", "iPhoneSE-
iPhoneSE", "iPhone7-iPhone7", "iPhone7Plus-iPhone7Plus", "iPad611-
iPad611", "iPad612-iPad612", "iPad71-iPad71", "iPad72-iPad72",
"iPad73-iPad73", "iPad74-iPad74", "iPhone8-iPhone8", "iPhone8Plus-
iPhone8Plus", "iPhoneX-iPhoneX"],
    "screenshotUrls":
["http://is1.mzstatic.com/image/thumb/Purple128/v4/df/85/fc/df85fc30-
6305-5736-9dba-0df527457f0d/source/392x696bb.jpg",
"http://is3.mzstatic.com/image/thumb/Purple118/v4/5c/b8/e5/5cb8e5f3-
a69e-55d5-35da-958106474204/source/392x696bb.jpg",
"http://is4.mzstatic.com/image/thumb/Purple118/v4/d7/e5/10/d7e51065-
c6d9-e175-6ab8-7dacfbcb10d09/source/392x696bb.jpg",
"http://is2.mzstatic.com/image/thumb/Purple128/v4/c2/37/0a/c2370a37-
2645-5f0f-b313-065314221590/source/392x696bb.jpg",
"http://is3.mzstatic.com/image/thumb/Purple118/v4/83/8c/99/838c99a4-
11a2-8fc0-f3e4-d64e4acb88e0/source/392x696bb.jpg"],
    "advisories": [],
    "trackCensoredName": "Words With Friends - Word Game",
    "trackViewUrl": "https://itunes.apple.com/us/app/words-
with-friends-word-game/id804379658?mt=8&uo=4",
    "contentAdvisoryRating": "4+",
    "languageCodesISO2A": ["EN", "FR", "DE", "IT", "PT", "ES"],
    "fileSizeBytes": "177342464",
    "sellerUrl": "https://zynga.com/games/words-friends",
    "averageUserRatingForCurrentVersion": 3.5,
    "userRatingCountForCurrentVersion": 409,
    "trackContentRating": "4+",
    "minimumOsVersion": "9.0",
    "currency": "USD",
    "wrapperType": "software",
    "version": "5.65",

```

```

"artistId": 295913422,
"artistName": "Zynga Inc.",
"genres": ["Games", "Board", "Entertainment", "Word"],
"price": 0.00,
"description": "The World's Most Popular Mobile Word Game now has even more ways
to play! Try new rewards and boosts to help you play smarter and faster!\n\nWords With Friends is the
fun, free word game that lets you connect with friends and family while expanding your vocabulary and
sharpening your wit. Test your brain power and spell your way to victory. \n\nMay the Best Friend
Win.™ \n\nGame Features: \n\n•\tWEEKLY CHALLENGE: Play the weekly challenge and earn fun,
themed badges\n\n•\tSOLO PLAY: Play offline and sharpen your skills with solo play\n\n•\tHINDSIGHT:
Uncover the best word you could have played after each turn \n\n•\tWORD RADAR: Heat map reveals all
possible moves before you take a turn\n\n•\tPROFILE FRAMES: Show off your unique style with custom
profile frames \n\n•\tSMART MATCH: Connect with players of a similar skill with Smart Match\n\n•\t6
LANGUAGES: Choose to play in Spanish, French, German, Italian, Brazilian Portuguese and British
English\n\n•\tiMESSAGE: Available on iMessage! Invite friends using your phone's contact list and make
moves directly in Messages\n\n*Play Words With Friends without third party ads between moves if you
previously purchased Words With Friends Pro on iOS or any other mobile device. Be sure to login with
Facebook or the email account you used to make the previous purchase to continue to play without
third party ads between moves. Games played in App Store for iMessage are separate from games on
your iPhone, iPad, Computer, and Apple Watch and will be separate games from those on other devices.
\n\nAlready an accomplished Wordie? Like us on Facebook or follow us on Twitter to stay up-to-date on
game news, contests, polls and
more.\n\nhttps://www.facebook.com/WordsWithFriends\n\nhttps://twitter.com/WordsWFriends
\n\nhttps://instagram.com/wordswithfriends \n\nWe love hearing from you! Have a question or a
suggestion? \n\nAsk questions here: http://zynga.tm/d01Tx \n\nShare suggestions here:
http://zynga.tm/r02dY \n\nThank you for playing Words With Friends!\n\nAdditional Information:\n\nThe
game is free to play; however, in-app purchases are available for additional content and in-game
currency. In-app purchases range from $0.99 to $99.99 USD.",
"bundleId": "com.newtoyinc.NewWordsWithFriendsFree",
"trackId": 804379658,
"trackName": "Words With Friends - Word Game",
"isVppDeviceBasedLicensingEnabled": true,
"primaryGenreName": "Games",
"releaseDate": "2014-10-09T03:36:24Z",
"primaryGenreId": 6014,
"sellerName": "Zynga Mobile",
"formattedPrice": "Free",
"genreIds": ["6014", "7004", "6016", "7019"],
"currentVersionReleaseDate": "2017-10-09T18:26:31Z",
"releaseNotes": "Dear Wordies,\n\nWe've updated our latest
version with bug fixes and improvements!\n\nMay The Best Friend
Win\n\nWords With Friends",
"averageUserRating": 4.5,

```

```
        "userRatingCount": 166376
    }
}
```

The domain is found in the sellerUrl field.

Appendix B: Proposed AppStore API

This API adheres to many of the conventions of RESTful APIs. The base protocol used for communication is HTTPS, and JSON is used to represent the body of requests and responses. Requests should be made with an HTTP header of “Accept: application/json” and “Content-Type: application/json” to indicate that the body of the request will be JSON and that JSON is expected in return.

Almost all fields are optional at a technical level, however app stores may mandate the presence of certain fields as a business requirement. Both app stores and end users must gracefully deal with the presence of unexpected or unknown fields. Breaking changes are restricted to major versions of this specification (1.x, 2.x, etc.).

HTTP status codes are used by the app store to express the status of requests made by the bidder:

Code	Name	Description
200	OK	The request was successful.
400	Bad Request	The request could not be interpreted successfully.
401	Unauthorized	The request did not contain correct authentication information.
404	Not Found	The resource does not exist.
429	Too Many Requests	The bidder has exceeded the rate limit set by the supplier and must wait before trying again.
500	Internal Service Error	The supplier has encountered technical difficulties.

Endpoints

Users will interact with the Appstore API by making HTTP calls to specific endpoints. App stores will specify a base URL (denoted using the {baseUrl} placeholder in this document. In order to signal the “App ID” (Bundle ID or Application package), we will use the placeholder {appId} in this document. All endpoints are relative to this base URL (indicated with the {baseUrl} placeholder throughout this document), proceeded with the version of the specification in the form of “v#. #”. For example, an app store may define its base URL as

“https://api.appstore.com/management”, in which case, assuming version 1.0 of this specification the api endpoint will be reached at “https://api.appstore.com/management/v1.0/app/com.my.bundle”.

Endpoint	Methods supported	Description
/search/	POST	Returns the data (result object) for a given search/filters (filter object)
/app/{appId}/	GET	Returns the data for a given app (app object)

Authentication

HTTP Basic authentication is used. App stores will provide users with a username and password. This username and password are combined with a colon (username:password) and base64 encoded. The result is used in the Authorization header in all calls the bidder makes to the API, e.g:

Authorization: Basic <base64 encoded value of "username:password">

Resource representations

Resources are represented using JSON.

Filter object

A filter resource is an object containing details of a search which a user may use to find apps in an app store. The object is sent via HTTP POST to the /search/ endpoint.

Attribute	Type	Description
lastmod	string	Optional. The apps returned can be filtered to those with information last modified at or later than a date/time by including the property lastmod and a date/time in the format of ISO 8601 (using the profile defined by W3C). For example: 2016-08-14T17:51:54Z

appids	array of string	Recommended. List of “app IDs” (Bundle ID or Application package) to search for. App stores may define that this property is required. App stores may define a maximum of IDs that can be passed.
offset	integer	Optional. Number that denotes the start of the result set of a search.
limit	integer	Optional. Number that denotes the maximum amount of results for a search.

Result object

A collection of app objects as a result of a search.

Attribute	Type	Description
ts	string	Optional. The current date/time at the app store (ISO 8601).
count	integer	Required. The number of apps in this collection.
total	integer	Optional. The total number of apps found for this search. If this property is not present, it is assumed that all apps are returned with this result.
apps	object array	Optional. An array of app objects. Property can be missing if no apps were found.

App object

An object containing metadata about a specific app.

Attribute	Type	Description
lastmod	string	Required. Date/time of last change of the app details (ISO 8601).

id	integer	Required. Bundle ID or Application package of the app.
name	string	Optional. Public name of the app.
sellerURI	string	Required. URI of the seller (app owner) where to find further details about the seller. The domain from this property can be used in order to find location of ads.txt/ads.cert files.
version	string	Optional. Public version number of the app, usually something like 1.2.345