

# Analytics for Digital Assistants Whitepaper

[View in implementation help](#)

## Introduction

With recent advances in cloud computing, machine learning and natural language processing, digital assistants are moving out of the dark ages of "clippy" and becoming part of a everyday life. Consumers are now starting to talk to their devices and expecting them to listen, understand and respond in very natural, human like ways. Phrases like "Alexa, turn on the family room lights", "Okay Google whats the weather like outside".

As these platforms become more established, brands can present their services to consumers in these same realistic and lifelike ways. For example consumers can ask things like.

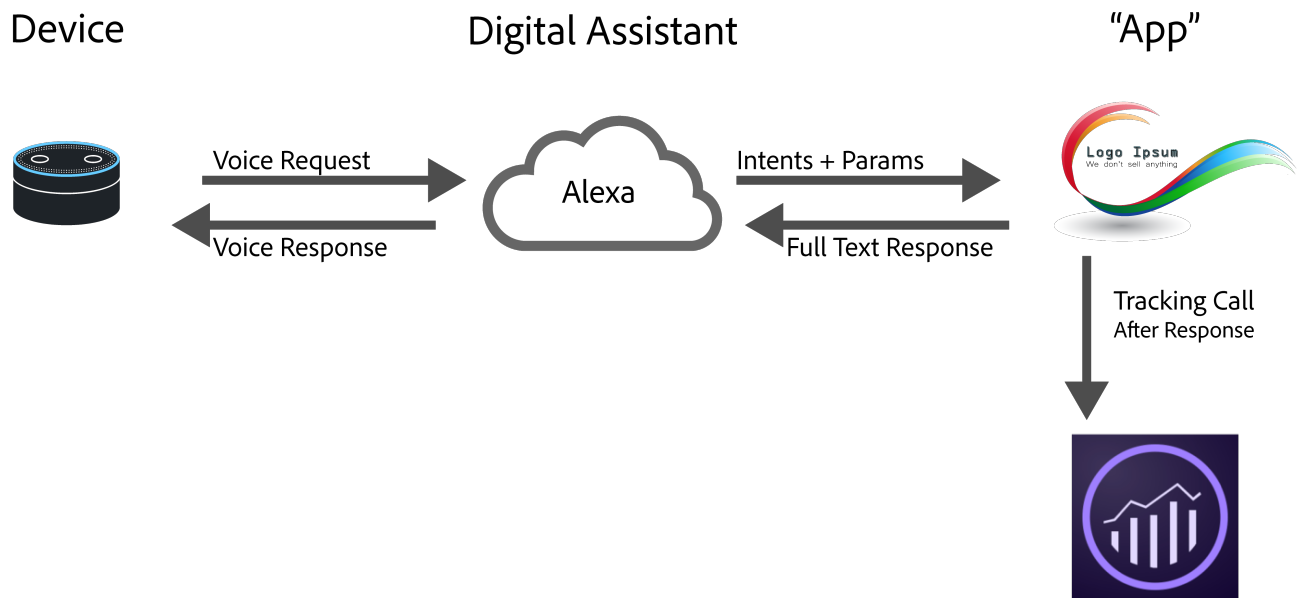
"Alexa, ask my car when it needs an oil change"

"Cortana, what is the balance of my checking account"

"Siri, Send John \$20 for dinner last night from my banking app"

This whitepaper will give you an overview of how best to use the Adobe Analytics Cloud to measure and optimize these types of experiences.

## Digital Experience Architecture Overview



Most Digital Assistants today follow a similar high-level architecture.

1. **Device** - There is a device (like an Amazon Echo, or a phone) with a microphone that allows the user to ask a question.
2. **Digital Assistant** - That device interacts with the service that powers the digital assistant. This service is where a lot of the "magic" happens. It is where the speech is converted into machine understandable intents and the details of the request are parsed out. Once the users intent is understood will pass the intent and details of the request to the app that will handle the request.
3. **"App"** - The app can either be an app on the phone or a voice app. The app is responsible for responding to the request. It responds to the digital assistant and the digital assistant then responds to the user.

## Where to implement Analytics

One of the best places to implement Analytics is in the "App". The app is what receives the intent and the details about the intent from the digital assistant and decides how to respond.

There are two times during the lifecycle of a request that can be helpful to call the Analytics Cloud.

1. When the request is sent to the "App" – If you need additional context about the user before you respond to the request you will want to reach out the Audience Manager capability to get the segments that they belong to.
2. After the response is returned from the "App" – If you are just interested in recording what happened with the customer for future optimization, send a request to Adobe Analytics after the response has been returned. This way you have the full context of what the request was and how the system responded.

## Analytics Implementation for Digital Assistants

### New Installs

For some of the digital assistants you will get a notification when someone installs the skill. This is especially the case when there is authentication involved. At this time you should send Adobe an Install event by setting the context data to a.InstallEvent=1. Note this isn't available on all platforms but is helpful when it is present for looking at retention. Below code sample, sends in Install, Install Date, and AppID.

--Code Sample --

```
GET /b/ss/[rsid]/0?vid=[UserID]&c.a.InstallEvent=1&c.a.InstallDate=2017-04-24&c.a.AppID=Spooify1.0&c.OSType=Alexa&pageName=install HTTP/1.1
Host: [prefix].sc.omtrdc.net
Cache-Control: no-cache
```

### Multiple Assistants or Multiple Apps

It is likely that you will develop "Apps" for multiple platforms. It is a best practice to include an app id with each request. This can be set in the a.AppID context data. Usually you will want to follow the format of [AppName] [BundleVersion] for example BigMac for Alexa 1.2

-- Code Sample --

```
GET /b/ss/[rsid]/0?vid=[UserID]&c.a.AppID=Spooify1.0&c.a.Launches=1&c.Product=AmazonEcho&c.OSType=Alexa&pageName=install HTTP/1.1
Host: [prefix].sc.omtrdc.net
Cache-Control: no-cache
```

```
GET /b/ss/[rsid]/0?vid=[UserID]&c.a.AppID=Spooify2.0&c.a.Launches=1&c.Product=GoogleHome&c.OSType=Android&pageName=install HTTP/1.1
Host: [prefix].sc.omtrdc.net
Cache-Control: no-cache
```

### User/Visitor Identification

The Analytics Cloud uses a visitor id to tie interactions across time to the same person. Most of the digital assistants will return a UserID that you can use to keep the activity for different users separate in the service. In most cases this is what you should pass in as the visitor ID. Some platforms return a userID that is longer than 100 characters that the Analytics limit allows. If that is the case then we advise that you hash the userID to a fixed length value using a standard hashing algorithm such as MD5 or Sha1.

While you can use the Marketing Cloud visitor ID service for this it will only provide value if you are trying to map visitor IDs across different devices (e.g. Web to Digital Assistant). If your "App" is a mobile app (e.g. a Deep link) you should use the SDK as is and just send the information along. The UserID can be attached to the visitor ID service using the setCustomerID method, allow for better device stitching. However, if the "App" is a service then you will want to use the UserID provided by the service as the visitorID as well as setting it in the setCustomerID. That will allow you to see how people are using the digital assistant over time.

--Code Sample --

```
GET /b/ss/[rsid]/0?vid=[UserID]&pageName=[intent] HTTP/1.1
Host: [prefix].sc.omtrdc.net
Cache-Control: no-cache
```

### Sessions

Because digital assistants are conversational they often have the concept of a session. Here is an Example:

Consumer: "Ok Google, call a cab for me"

Google: "Sure, what time would you like?"

Consumer: "8:30pm"

Google: "Sounds good, the Driver will be by at 8:30pm"

These sessions are important to keep in context. They help collect more details and make the digital assistants more natural.

When implementing Analytics on a conversation you there are two things you will want to do when a new session is started.

1. Reach out to Audience Manager – To get the relevant segments that a user is a part of so that you can customize the response. (E.g. this person currently qualifies for the multi-channel discount)
2. Send in a new session or Launch event – When you send the first response to Analytics include a launch event. This can usually be sent by setting context data of a.LaunchEvent=1

--Code Sample for Launch --

```
GET /b/ss/[rsid]/0?vid=[UserID]&c.a.LaunchEvent=1&c.Intent=[intent]&pageName=[intent] HTTP/1.1
Host: [prefix].sc.omtrdc.net
Cache-Control: no-cache
```

## Intents

Each of the digital assistants has algorithms that will detect intents and then pass the intent down to the "App" so the app knows what to do. These intents are a succinct representation of the request. For example if a user says.

"Siri, Send John \$20 for dinner last night from my banking app" the intent might be something like "sendMoney".

By sending in each of these requests as an event you will be able to run pathing reports on each of the intents for conversational apps. You will also want to make sure the "App" will be able to handle requests without an intent. We recommend passing in "No Intent Specified" as opposed to leaving the value blank.

-- Code Sample --

```
GET /b/ss/[rsid]/0?vid=[UserID]&c.a.AppID=Penmo1.0&c.a.LaunchEvent=1&c.Intent=SendPayment&pageName=[intent] HTTP/1.1
Host: [prefix].sc.omtrdc.net
Cache-Control: no-cache
```

or

```
GET /b/ss/[rsid]/0?vid=[UserID]&c.a.AppID=Penmo1.0&c.a.LaunchEvent=1&c.Intent=No_Intent_Specified&pageName=[intent] HTTP/1.1
Host: [prefix].sc.omtrdc.net
Cache-Control: no-cache
```

## Parameters/Slots/Entities

In addition to the intent the digital assistant will often have a set of key value pairs that give the details of the intent. These are called slots, entities or parameters. For example

"Siri, Send John \$20 for dinner last night from my banking app" would have the following parameters.

Who = John

Amount = \$20

Why = Dinner

There is usually a finite number of these with your app. To track these in Analytics you will want to send them into context data and then map each of the parameters to an eVar.

--Code Sample --

```
GET
/b/ss/[rsid]/0?vid=[UserID]&c.a.AppID=Penmo1.0=1&c.a.LaunchEvent=1&c.Intent=SendPayment&c.Amount=20.00&c.Reason=Dinner&c.ReceivingPerson=John&c.Intent=SendPayment&pageName=[intent] HTTP/1.1
Host: [prefix].sc.omtrdc.net
Cache-Control: no-cache
```

## Success

While not all apps will be revenue generating it is important to think about what success looks like and include some measure for that. Adobe Analytics can measure, revenue, ad-impressions and other forms of success along with the user behavior.

## Error States

Sometimes the Digital Assistant will provide the app with inputs that it doesn't know how to handle. For example.

"Siri, Send John 20 bags of coal for dinner last night from my banking app"

When this happens the app should ask for clarification. Additionally when it is responding to a request like this you should send Analytics an event that indicates the App has an error state along with an event that specifies what type of error occurred.

Be sure to include errors where the inputs are not correct and errors where the "App" had a problem.

--Code Sample --

```
GET /b/ss/[rsid]/0/?vid=[UserID]&c.a.AppID=Penmo1.0&c.Error=1&c.ErrorName=InvalidCurrency&pageName=[intent] HTTP/1.1
Host: [prefix].sc.omtrdc.net
Cache-Control: no-cache
```

## Device Capabilities

While most of the platforms don't expose the actual device that the user spoke too. They do expose the capabilities of the device as the available interfaces (e.g. Audio, Screen, Video, etc). This is useful information because it defines the types of content that can be used when interacting with your users. When measuring the interfaces it is best to concatenate them (in alphabetical order).

e.g. :Audio:Camera:Screen:Video:

Notice the trailing and leading colons. These help when creating segments (e.g. give me all interactions with :Audio: capabilities).

Amazon Capabilities: <https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit/docs/alexa-skills-kit-interface-reference>

Google Capabilities: <https://developers.google.com/actions/assistant/surface-capabilities>

```
GET /b/ss/[rsid]/0/?vid=[UserID]&c.a.AppID=Penmo1.0&c.Capabilities=%3AAudio%3ACamera%3AScreen%3AVideo%3A&pageName=[intent] HTTP/1.1
Host: [prefix].sc.omtrdc.net
Cache-Control: no-cache
```

## Analytics Reporting for Digital Assistants

Once your Digital Assistant App is implemented you can use the full power of Adobe Analytics with it. Below are just a few examples of the things you can do with Analytics.

### Monitoring Intents

Most Apps have several intents and different things you can do. You could easily use Analysis Workspace to keep track of the top intents by instances and by users

--- Image of Intents ---

This lets you see which features are being used most often and can give you a view into the adoption of new features.

### Requests with Errors

You will be able to monitor errors to see if there are common places where users are getting having problems.

-- Image of Errors --

### Flow Between Intents

One of the most powerful things to do is look at the flow of intents. This is helpful in two ways. First, you can look within a session for how people flow between intents in a conversation. Second, you can look at how people flow between intents over longer timeframes to see how their usage

of the "App" is evolving.

## ---Example---

### Pre-installed app

Person	Device Response	Action / Intent	Get Request	Analytics Data
Play Spoofify	"okay playing Spoofify"	Play	GET /b/ss/[rsid]/0?vid=[UserID]&c. a.AppID=Spoofify1.0&c.a.La unchEvent=1&c.Intent=Play &pageName=PlayApp HTTP /1.1 Host: [prefix].sc.omtrdc.net Cache-Control: no-cache	<ul style="list-style-type: none"><li>• Visitor ID</li><li>• App Version</li><li>• # of Launches</li><li>• Intent</li><li>• Response</li></ul>
Change song	"okay what song do you want?"	ChangeSong	GET /b/ss/[rsid]/0?vid=[UserID]&c. a.AppID=Spoofify1.0&c.Inten t=ChangeSong&pageName= AskForSong HTTP/1.1 Host: [prefix].sc.omtrdc.net Cache-Control: no-cache	<ul style="list-style-type: none"><li>• Visitor ID</li><li>• App Version</li><li>• Intent</li><li>• *blank playlist</li><li>• Response</li></ul>
Play "My Heart Will Go On" by Celine Dion	"okay playing 'My Heart Will Go On' by Celine Dion"	ChangeSong	GET /b/ss/[rsid]/0?vid=[UserID]&c. a.AppID=Spoofify1.0&c.Inten t=ChangePlaylist&pageNam e= ActionPlaySong&c.SongID=[ 012345] HTTP/1.1 Host: [prefix].sc.omtrdc.net Cache-Control: no-cache	<ul style="list-style-type: none"><li>• Visitor ID</li><li>• App Version</li><li>• Intent</li><li>• Song ID</li><li>• Response</li></ul>
Change playlist	"okay what playlist do you want?"	ChangePlaylist	GET /b/ss/[rsid]/0?vid=[UserID]&c. a.AppID=Spoofify1.0&c.Inten t=ChangePlaylist&pageNam e= AskForPlaylist HTTP/1.1 Host: [prefix].sc.omtrdc.net Cache-Control: no-cache	<ul style="list-style-type: none"><li>• Visitor ID</li><li>• App Version</li><li>• Intent</li><li>• *blank playlist</li><li>• Response</li></ul>
Play Usher	"okay playing Usher"	ChangePlaylist	GET /b/ss/[rsid]/0?vid=[UserID]&c. a.AppID=Spoofify1.0&c.Inten t=ChangePlaylist&pageNam e= ActionPlayPlaylist&c.Playlist =Usher HTTP/1.1 Host: [prefix].sc.omtrdc.net Cache-Control: no-cache	<ul style="list-style-type: none"><li>• Visitor ID</li><li>• App Version</li><li>• Intent</li><li>• Playlist</li><li>• Response</li></ul>
Turn music off	*no response, music turns off	Off	GET /b/ss/[rsid]/0?vid=[UserID]&c. a.AppID=Spoofify1.0&c.Inten t=Off&pageName=TurnsOff Music HTTP/1.1 Host: [prefix].sc.omtrdc.net Cache-Control: no-cache	<ul style="list-style-type: none"><li>• Visitor ID</li><li>• App Version</li><li>• Intent</li><li>• Response</li></ul>

Requires user to install app

Person	Device Response	Action / Intent	Get Request	Analytics Data
Install Spoofify	*no response	Install	GET /b/ss/[rsid]/0?vid=[UserID]&am;c.a.InstallEvent=1&c.a.InstallDate=2017-04-24&c.a.AppID=Spoofify1.0&c.OSType=Alexa&c.Intent=Install&pageName=Install HTTP/1.1 Host: [prefix].sc.omtrdc.net Cache-Control: no-cache	<ul style="list-style-type: none"> <li>• Visitor ID</li> <li>• Date</li> <li>• Intent</li> <li>• OS Version</li> <li>• Response</li> </ul>
Play Spoofify	"okay playing Spoofify"	Play	GET /b/ss/[rsid]/0?vid=[UserID]&c.a.AppID=Spoofify1.0&c.a.LaunchEvent=1&c.Intent=Play&pageName=PlayApp HTTP/1.1 Host: [prefix].sc.omtrdc.net Cache-Control: no-cache	<ul style="list-style-type: none"> <li>• Visitor ID</li> <li>• App Version</li> <li>• # of Launches</li> <li>• Intent</li> <li>• Response</li> </ul>
Change song	"okay what song do you want?"	ChangeSong	GET /b/ss/[rsid]/0?vid=[UserID]&c.a.AppID=Spoofify1.0&c.Intent=ChangeSong&pageName=AskForSong HTTP/1.1 Host: [prefix].sc.omtrdc.net Cache-Control: no-cache	<ul style="list-style-type: none"> <li>• Visitor ID</li> <li>• App Version</li> <li>• Intent</li> <li>• *blank playlist</li> <li>• Response</li> </ul>
Play "My Heart Will Go On" by Celine Dion	"okay playing 'My Heart Will Go On' by Celine Dion"	ChangeSong	GET /b/ss/[rsid]/0?vid=[UserID]&c.a.AppID=Spoofify1.0&c.Intent=ChangePlaylist&pageName=ActionPlaySong&c.SongID=[012345] HTTP/1.1 Host: [prefix].sc.omtrdc.net Cache-Control: no-cache	<ul style="list-style-type: none"> <li>• Visitor ID</li> <li>• App Version</li> <li>• Intent</li> <li>• Song ID</li> <li>• Response</li> </ul>
Change playlist	"okay what playlist do you want?"	ChangePlaylist	GET /b/ss/[rsid]/0?vid=[UserID]&c.a.AppID=Spoofify1.0&c.Intent=ChangePlaylist&pageName=AskForPlaylist HTTP/1.1 Host: [prefix].sc.omtrdc.net Cache-Control: no-cache	<ul style="list-style-type: none"> <li>• Visitor ID</li> <li>• App Version</li> <li>• Intent</li> <li>• *blank playlist</li> <li>• Response</li> </ul>
Play Usher	"okay playing Usher"	ChangePlaylist	GET /b/ss/[rsid]/0?vid=[UserID]&c.a.AppID=Spoofify1.0&c.Intent=ChangePlaylist&pageName=ActionPlayPlaylist&c.Playlist=Usher HTTP/1.1 Host: [prefix].sc.omtrdc.net Cache-Control: no-cache	<ul style="list-style-type: none"> <li>• Visitor ID</li> <li>• App Version</li> <li>• Intent</li> <li>• Playlist</li> <li>• Response</li> </ul>
Turn music off	*no response, music turns off	Off	GET /b/ss/[rsid]/0?vid=[UserID]&c.a.AppID=Spoofify1.0&c.Intent=Off&pageName=TurnsOffMusic HTTP/1.1 Host: [prefix].sc.omtrdc.net Cache-Control: no-cache	<ul style="list-style-type: none"> <li>• Visitor ID</li> <li>• App Version</li> <li>• Intent</li> <li>• Response</li> </ul>

## Resources

- [Adobe Analytics Data Insertion Docs](#)
- [Alexa Developer Docs](#)
- [Alexa Skills Kit](#)
- [Google Assistant Developer Docs](#)
- [Cortana Developer Center](#)
- [SiriKit](#)