Catagory	Name	URL/API; change the parameter in the {} with the right value	Note	Parameters	Description	*Sample JSON (it may returns more than one)*	
WIFI	Wifi settings	https://pitangui.amazon.com/api/wifi/configs	----	None	Wifi settings used by the alexa device including the passwords.	*{ "deviceSerialNumber":null, "deviceType":null, "preSharedKey":"********1" "securityMethod":"WPA_PSK", "ssid": "wifi_name" }*	
	Current wifi in use	https://pitangui.amazon.com/api/device-wifi-details?deviceSerialNumber={***DeviceSerialNumber***}&deviceType={***deviceType***}	----	***DeviceSerialNumber:*** Serial number of the device ***deviceType***: device type. These parameters can be found using the above API	Device wifi detail	*{ "deviceSerialNumber": "serial number of the device", "deviceType": "device type (echo)", "essid": "wifi_name", "macAddress": "MacAdress of the current device (echo)" }*	
Devices	Devices	https://pitangui.amazon.com/api/devices/device	----	None	Alexa enabled devices such as Echo, Dot...	*"devices": [{ "accountName": "name's Echo", ... }, "accountName": "names's Amazon Apps",.... }*	
	Device preference	https://pitangui.amazon.com/api/device-preferences	----	None	list of alexa enabled devices with their preference and settings. This include serial number, device type, time zone, postal code and so on.	*{ "deviceSerialNumber": "device serial num", "locale": "locale", "postalCode": "registered postal code", "timeZoneId": "regirtered time zone" }*	
	Connected devices	https://pitangui.amazon.com/api/phoenix	----	None	Detected compatible smart devices that alexa can see in the smart Home	*{ "manufacturerName":"Nest", "modelName":"Nest Thermostat", "version":"version", "entityId":"id", "isEnabled":true }*	
Cards	Active cards	https://pitangui.amazon.com/api/cards?limit=50&beforeCreationTime=	This API returns limited number of active cards, to get the next active cards change the parameter ***beforeCreationTime*** with a value with value ***nextQueryTime*** from the returned JSON data. This way, we can get all active data only by changing this parameter	***limit***: number of card to return ***beforeCreationTime***: return after this time. Start by leaving this parameter null, it will return the first active cards.	These are active cards or conversation between alexa and the user or with any other device.	*{ "cardType": "TextCard or wheather card...", "creationTimestamp": 1484539461678, "descriptiveText": ["What Alexa answered"], "playbackAudioAction": { "mainText": "text what Alexa heard", "url": "URL of the voice file on the cloud" }, "title": "Do you know the muffin man?" },*	
	Individual cards	https://pitangui.amazon.com/api/cards/{***cardId***}	cardId: the id of specific card from the list of cards returned using the above API	None	Individual card from the list of cards from the above API, the above API returns around 21 maximum cards at once.	*Same as above, but single card.*	
	Individual voices	https://pitangui.amazon.com/{***url***}	url: get this value from the above API, there is url value on the returned JSON	None	Audio data, if availbale, of individual card.	*bytes*	
Lists	To-do lists	https://pitangui.amazon.com/api/todos?type=TASK&size=100&complete=true|false	----	***type:*** the type of list to return, in this case it is TASK ***size:*** number of lists to return, default 100 ***complete:*** return all lists	To do list created by the user	*{ "createdDate": 1480350314486, "lastupdatedDate" : 1480350314486, "customerId": "customer id", "originalAudioId": "URL of the voice file on the cloud", "text": "Re-install the app" "type": "TASK", }*	
	Shopping lists	https://pitangui.amazon.com/api/todos?type=SHOPPING_ITEM&size=100&complete=true|false	----	***type:*** the type of list to return, in this case it is SHOPPING_ITEM ***size:*** number of lists to return, default 100 ***complete:*** return all lists	Shopping lists created by the user.	*{ "createdDate": 1480350314486, "lastupdatedDate" : 1480350314486, "customerId": "customer id", "originalAudioId": "URL of the voice file on the cloud", "text": "Buy Battery", "type": "SHOPPING_ITEM", }*	
	Named list	https://alexa.amazon.com/api/namedLists	Newly added API that contain user created lists combining the shopping, To-Do, and custom lists.				
	Specific list (Items)	https://alexa.amazon.com/api/***namedLists***/{***itemId***}/items?startTime=&endTime=&completed={***true/false***}&listIds={***listIds***}	***itemId*** and ***listId*** values are available in the above API, replace with approprait value. Also, make completed parameter true to return only completed lists. Note: change equal (=) sign in listIds with (%3d), its hex value.				
User settings	Bluetooth	https://pitangui.amazon.com/api/bluetooth	----	none	Blutooth setting of the device including list of paired devices	*{ "deviceSerialNumber": "serial number", "friendlyName": "device name" "pairedDeviceList": [{ "address":"Mac address", "friendlyName": "device name", }*	
	Accounts	https://pitangui.amazon.com/api/authentication	----	none	account information of the user including email, full name and customer id.	*{ "authenticated": true, "customerEmail": "user email", "customerId": "**2T**M**OXJ**", "customerName": "name" }*	
	House hold information	https://pitangui.amazon.com/api/household	----	none	user information including full name and email	*{ "email": "user email", "firstName": "name", "fullName": "full name", "id": "**2T**M**OXJ**", "role": "ADULT" }*	
	Notifications	https://pitangui.amazon.com/api/notifications	----	none	timer and alarm lists that a user set	*{ "alarmTime": 1499958000000, "createdDate": 1499931193511, "deviceSerialNumber": "device serial number", "status": "ON", "type": "Alarm" }*	
User Activities	All activities	https://pitangui.amazon.com/api/activities?startTime=**{startDate}**&endTime=&size=50&offset=-1	This API returns the last 50 activities a user command alexa. To get more activities, we should change the value of ***startTime*** on the API with the value of ***startDate*** on the returned JSON data.	***startTime:*** return activities starting from this time, it is in epoch time format. Leaving this parameter null will return the latest 50 activities. ***endTime:*** return activity upto this time, this also epoch time. this can be null ***size:*** *number of activities to return, maximum 50 activities.* ***offset:*** -1, 0 or 1. -1 iterates the activity backward.	Activity between the user and the device. This is history of user commands.	*{ "activityStatus": "SUCCESS", "creationTimestamp": 1511793618587 (EPOCH), "registeredCustomerId": "**2T**M**OXJ**", "description":{ "summary":"alexa heared", }, "utteranceId": "Voice file URL to the cloud" }*	
	Individual activities	https://pitangui.amazon.com/api/activity-dialog-items?activityKey={***id***}	get the value of id, the variable in the parenthesis, from the each activity from the above API to get individual dialog, this contains the whole process of the command,	***activityKey:*** assign this parameter with the individual acvtivity id that is returned using the above API. Single request using the above API returns 50 individual activity, if available and size is not assigned less that 50.	Individual activity from the returned 50 activities using the above API	*"activityDialogItems": [{ activityItemData:{"asrText":"send a message"} "itemType": "ASR", //automatic speach recognition "timestamp": 1511336576233, "utteranceId":"URL of the audio to the cloud" }, { "activityItemData": "Intents or skills", "itemType": "NLU", //natural language understanding "timestamp": 1511336576233, "utteranceId":"URL of the audio to the cloud" }*	

	Voice	https://pitangui.amazon.com/api/utterance/audio/data?id={***utteranceId***} .	utteranceId is a value available in each individual activity of the above returned JSON values, and contains voice command of the user. To get the voice data from these activities, append the utteranceId from these individual activities in to this API.	***utteranceId:*** a value available in each individual activity from the above two APIs. In the activity dialog, there maybe more than one voice data because sometimes alexa asks a series of questions to clarify or to get more detail. in this case, in one dialog activity we will get more than one audio data. Use this method to get all the available audios.	Audio data of individual activities.	*.wav file. Bytes*	
Calling and messaging	account detail	https://alexa-comms-mobile-service-na.amazon.com/accounts/	----	None	Account info including full name, phone number, and communication id.	*{ "commsId":"amzn1.comms.id.person.amzn1~amzn1.account.**P3Q***GF***D**S3**J7*U****", "directedId":"amzn1.account.**P3Q***GF***D**S3**J7*U****", "phoneCountryCode": "82", "phoneNumber": "phone number", "firstName": "First name of the user", "lastName": "last name", }*	
	user detail	https://alexa.amazon.com/api/users/me	----	none	User detail such as email, name, country and id.	*{ "countryOfResidence": "KR", "email": "email@something.com", "fullName":"user full name", "id": "***TR**6QOX****", }*	
	contacts	https://alexa-mobile-service-na-preview.amazon.com/users/{***commsId***}/contacts	***commsId:*** get commsId from account detail API	None	Contact book of the user. The contacts in this list should have adevice of their own.	*{ "name": {""full name"}, "emails": null, "number": "phone number", "commsId": ["amzn1.comms.id.person.amzn1~amzn1.account.***3QR6**FO24***VS3***7NUI****"], }*	
	conversations	https://alexa-mobile-service-na-preview.amazon.com/users/{***commsId***}/conversations	***commsId***: get commsId from account detail API	None	list of Conversation the user exchanged with other contacts	*"conversations": [{ "conversationId":"amzn1.comms.messaging.id.conversation~x*Nil**Ri-ce***zU-**3aBz****", "firstVisibleMessageId": 1, "lastMessageId": 14, "sendAsCommsId":"amzn1.comms.id.person.amzn1~amzn1.account.***3QR6**FO24***VS3***7NUI****" }*	
	individual conversations	*https://alexa-comms-mobile-service-na.amazon.com/users/{**commsId**} /conversations/ {**conversationId**}/messages? **startId**={}&**count**={}*	***commsId:*** get commsId from account detail JSON, or from the conversation JSON ***conversationId:*** get from conversation API, each conversation has its own conversationId.	'startId': start of message Id, starts from 1. 'count': number of messages to return 'sort': default = ascending.	Single or grouped conversation in one thread	*"messages": [{ "conversationId":"amzn1.comms.messaging.id.conversation~x*Nil**Ri-ce***zU-**3aBz****", "messageId": 14, "time": "2017-11-22T07:43:15.468Z", "sender":"amzn1.comms.id.person.amzn1~amzn1.account.***3QR6**FO24***VS3***7NUI****", "type": "message/text", "payload": { "text":"text message" } }*	
	User identity	https://alexa-comms-mobile-service.amazon.com/users//{commsId}/identities	User identity detail including commsId and homeGroupId			{ "commsId": "" "homeGroupId": "" }	
	Home group contacts	https://alexa-comms-mobile-service.amazon.com/users/{homeGroupId }/contacts/	Contacts in the user's home group. Get ***homeGroupId*** from the above API return JSON.			Same as contacts but in the user's group.	