

MobiBox Users Aquisition API

Company	 MOBIBOX OUT OF THE BOX TECHNOLOGY MOBIBOX TECHNOLOGIES
Version	1.0
Release date	13/01/2023

i THE SPECIFICATIONS AND INFORMATION IN THIS MANUAL ARE SUBJECT TO CHANGE.
ALL INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE.
USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR TECHNICAL IMPLEMENTATION.

Table Of Content [↗](#)

- [Table Of Content](#)
- [Introduction](#)
- [API Request](#)
 - [DeviceInfo Attributes](#)
 - [Referrer Attributes](#)
 - [Request Attributes](#)
- [API Response](#)
- [Annex](#)
 - [Actions](#)
 - [Possible Returned Actions](#)
 - [NextAction Object](#)
 - [Fixed Attributes](#)
 - [Additional Dynamic Attributes Per Returned Action](#)
 - [Codes](#)
 - [Adding AFScript](#)
 - [Encrypt Request / Decrypt Response Functions](#)
 - [C#](#)
 - [PHP](#)
 - [Examples](#)
 - [Request Body](#)
 - [Responses](#)
 - [Error](#)
 - [Next Action is SendPin, ClicksFlow, ClickToSMS](#)
 - [Next Action is VerifyPin](#)
 - [Next Action is LoadURL](#)
 - [Next Action is SendSMS](#)
 - [NextAction is Close](#)

Introduction [↗](#)

- This API is designed to support various Mobile Operator Subscription flows
- Supported Subscription Flows are

- Pin Flow With MSISDN Entry
- SMS Flow With MSISDN Entry
- Click to SMS (SMS Without MSISDN Entry)
- 2 Clicks Flow
- Consent Gateway Flow (regardless if the Page is hosted at operator's side or client side)
- The API will guide the connected partner about what to do step by step, so as a partner integrating this API , you just send first action and then the API will tell what is the next step and the next action to send.
- The API uses AES Encryption/Decryption
- Each Partner will have a specific Key for Encryption/Decryption, a specific API endpoint and a specific AccessToken to be used in API Request's Header to Authenticate the requests
- All Requests are POST Requests with JSON object

API Request [↗](#)

- Url to Call : <https://endpoint/UsersAquisition/>
- “endpoint” value will be shared with you
- Header Authorization Parameter:
 - Key = AccessToken
 - Value = will be shared with you later
- Encryption/Decryption key = will be shared with you
- Encryption/Decryption IV = is the first 16 characters of the encryption key(e.g., encryption key is DVpqooCzLNOMUHFxAdRKF6iY6pWz0plq then IV will be DVpqooCzLNOMUHFx)
- Request Body should be encrypted using the encryption key,IV (see the encryption functions in Annex)
- Request Body is a JSON object consisting of 3 Main Nodes (DeviceInfo, Referrer, Request) and each one is a JSON object as well

DeviceInfo Attributes [↗](#)

Attribute	Type	Obligation	Description
LangCode	string	No	device's language (to return localized content if possible)

Referrer Attributes [↗](#)

is a JSON Object contains 1 attribute called “Affiliate” which is a JSON object as well contains the following Attributes

Attribute	Type	Obligation	Description
Campaign	string	Yes	Campaign Value we share with you
ClickID	string	No	Your clickid value
Pub_ID	string	No	Your publisher id

Aff_ID	string	No	Your affiliate id
extra	string	No	Extra parameter that can be used to send other values that are important to you
extra1	string	No	Same as “extra” parameter
Country	string	No	in alpha 2 format w ISO 3166-1 alpha-2 when provided we rely on it to open offers in that country, if not, we rely on user's detected country to get offers
firstPageButtonID	string	Yes	ID of the first page's button to be controled by our cybersecurity partners
secondPageButtonID	string	Yes	ID of the second page's button to be controled by our cybersecurity partners

Notes:

- ClickID, Pub_ID, Aff_ID, extra, extra1 can be sent back to you in your conversions pixel if needed
- the Buttons IDs values depend on the subscription flow
 - Pin Flow With MSISDN Entry (2 pages ... 1 for MSISDN entry and 1 for PIN entry .. so we need both values)
 - SMS Flow With MSISDN Entry (1 page for MSISDN entry so we need firstPageButtonID value only .. because after user clicks on that button .. there will be no second page to show and SMS composer should be opened to finish the subscription)
 - Click to SMS (1 page Without MSISDN Entry after user clicks on that button .. there will be no second page to show and SMS composer should be opened to finish the subscription)
 - 2 Clicks Flow (1 page Without MSISDN Entry .. after the first click we redirect user to Operator's Consent Gateway to do the second one and finish the subscription)
 - Consent Gateway Flow (you can send both attributes as an empty string .. cause here will be a redirection to an uncontrollable third party page)

Request Attributes [🔗](#)

Attribute	Type	Obligatory	Description
Action	integer	Yes	See the Actions in Annex. The first action you use is 1 This call (with action 1) should be done when user land on your page (on the load of the page before user sees anything) and it will returns to you what should the user see next
TransactionID	string	Yes	A unique id for each request. Any duplicated TransactionID will be rejected
SessionID	string	Yes/No	A unique value from our side to be sent in all requests. The first call you make you send it empty but after the response returns the SessionID , you should keep it and send it in all following requests (actions)
MSISDN	string	Yes/No	User's Phone Number. Obligatory with all MSISDN based subscription flows

PinCode	string	Yes/No	The pin that should be verified to subscribe. Obligatory in case of verify Pin action
---------	--------	--------	--

API Response [↗](#)

- API Response will be encrypted and you can decrypt it using the same encryption key (see the decryption function in Annex)
- Response is a JSON object consists of

Attribute	Type	Description
Error	integer	0 when all ok 1 when there is error and we should not continue with the flow (for example when MSISDN is not valid .. here we should not proceed to next step and user should correct the msisdn) 2 when user is already subscribed and we need to continue to NextAction
Description	string	The reason of error in case of Error=1
MessageToShow	string	A message to show to the user. when Error = 1 and this value is not an empty string you can show this message to user. examples of Messages : “Service not available in your country” .. “please enter a correct pin” .. etc
SessionID	string	Unique value from our side that should be sent in all actions
NextAction	JsonObject	represents what is the next action that should be taken. It contains the Action value and all related info (see Annex)
Payout	JsonObject	contains 2 attributes 1. Rate a. Type: double b. Values: 0.0 in case there is no conversion otherwise the conversion rate. 2. Currency a. Type: string b. Values: empty in case there is no conversion otherwise the currency name in 3 letters format w ISO 4217

Annex [↗](#)

Actions [↗](#)

Action	ID	Received By API	Description
Initiate	1	Yes	This is the first Action you send (sometimes we might ask for a different value depending on the integration) .. after that, the api will tell what is the next action to perform
SendPin	2	Yes	To send pincode to user

VerifyPin	3	Yes	To verify the pincode that user enters
LoadURL	4	No	Redirect to an URL to complete the subscription or to show content
SendSMS	5	No	You will receive this Action in the response whenever there should be SMS to be send from user's device .. so you open the sms composer
ClicksFlow	6	Yes	You will receive this Action in the response when there is 2 clicks flow or 1 click flow
Close	7	No	When you receive this action that means the subscription is completed or there is no need to continue .. so the connection should be terminated
ClickToSMS	8	Yes	You will receive this Action in the response when user should send sms without msisdn entry

Possible Returned Actions [🔗](#)

Received Action / ID	Possible Returned Actions / ID	Description
Initiate / 1	SendPin / 2	returned when the subscription is pin flow .. so when Initiate returns this Action .. user should see the MSISDN entry page with the disclaimers .. and when he clicks the first button you call the API with this Action (2) to send the pin to user's msisdn
	LoadURL / 4	returned when the subscription is Consent Gateway Flow. so when Initiate returns this Action .. you just redirect user to the returned URL and no more communication with the API
	ClicksFlow / 6	returned when the subscription is 1 or 2 clicks flow so when Initiate returns this Action .. you show the user a page contains the disclaimers and a button (like MSISDN Entry page but without the entry) .. and when he clicks the button you call the API with this Action (6) to get the next action
	ClickToSMS / 8	returned when the subscription is click to sms so when Initiate returns this Action .. you show the user a page contains the disclaimers and a button (like MSISDN Entry page but without the entry) .. and when he clicks the button you call the API with this Action (8) to get the SMS details
SendPin / 2	VerifyPin / 3	When SendPin action is received, the API will send the pin to user and returns this action ... so you show the user the PIN entry page .. and when he enters the pin and clicks the verify button .. you call the API with this Action(3) ..
	LoadURL / 4	returned when the Pin should be verified at Operator's Consent Gateway or user is already subscribed and should get content..... so you redirect him to the returned URL
	SendsMS / 5	returned when user should send sms to subscribe and not verifying pin so you should open the SMS composer with returned data so user can subscribe
VerifyPin / 3	LoadURL / 4	just redirect the user to the returned URL
ClicksFlow	LoadURL / 4	just redirect the user to the returned URL
ClickToSMS / 8	SendsMS / 5	returned when user should send sms to subscribe so you should open the SMS composer with returned data so user can subscribe

NextAction Object [🔗](#)

- this JSON Object contains 2 fixed attributes (always there) in addition to a set of dynamic attributes related to the returned Action

Fixed Attributes [🔗](#)

Attribute	Type	Description
Action	Integer	ID of the Next Action to be sent to Next API Call (check the 2 tables above)
Name	string	Name of the Next Action

Additional Dynamic Attributes Per Returned Action [🔗](#)

Returned Action	Attribute	Type	Description
SendPin / 2 ClicksFlow / 6 ClickToSMS / 8 (Please Check Note Below)	Disclaimers	Json Object	<p>this object Contains 3 disclaimers</p> <ul style="list-style-type: none">headerInfo : some operators ask to mention some text at the beginning of the landing pagemiddleInfo : some operators ask to mention some text between the data entry and the buttonfooterInfo : this is the main one that contains the pricing and other Operator's regulations <p>These 3 values are string with some HTML tags and styles, therefore we return them encoded (URLEncode) and you should decode them (URLDecode) before add them to your page</p>
	PrivacyPolicy	string	URL to show the Privacy Policy
	TermsAndConditions	string	URL to show the Terms and Conditions
	AFScript	string	cybersecurity script that you should load in your html page (Check Codes Section Below) in case it is not empty
LoadURL / 4	URL	string	the URL to redirect user to
VerifyPin / 3 (Please Check Note Below)	Disclaimers	Json Object	<p>some operators ask for some texts to be shown on the PIN verification page</p> <p>this object Contains 3 disclaimers</p> <ul style="list-style-type: none">headerInfo : to mention some text at the beginning of the landing pagemiddleInfo : to mention some text between the data entry and the buttonfooterInfo : to mention some text below the button <p>These 3 values are string with some HTML tags and styles, therefore we return them encoded (URLEncode) and you should decode them (URLDecode) before add them to your page</p>
	PinLengths	List<integer>	List of accepted pincode lengths

	PrivacyPolicy	string	URL to show the Privacy Policy
	TermsAndConditions	string	URL to show the Terms and Conditions
	AFScript	string	cybersecurity script that you should load in your html page (Check Codes Section Below) in case it is not empty
SendSMS / 5	Message	string	the text that should be sent as SMS (you will pre-fill the composer with this value)
	Destination	string	the receiver of the SMS (mainly it is a shortcode)

Note

sometimes TermsAndConditions, PrivacyPolicy and Disclaimers might not be in response at all .. it depends on the integration setup .. but AFScript will be always there even if empty

Codes [🔗](#)

Adding AFScript [🔗](#)

the following javascript code is what you should execute on your page to load our AFScript if it is not empty

```
1 var script = document.createElement("script");
2   script.type = "text/javascript";
3   script.innerHTML = response['AFScript'];
4   document.getElementsByTagName('head')[0].appendChild(script);
```

Encrypt Request / Decrypt Response Functions [🔗](#)

C# [🔗](#)

```
1 public static string Encrypt(string data, string _key)
2 {
3     string IV = _key.Substring(0, 16);
4     byte[] iv = Encoding.UTF8.GetBytes(IV);
5     byte[] array;
6     using (Aes aes = Aes.Create())
7     {
8         aes.Key = Encoding.UTF8.GetBytes(_key);
9         aes.IV = iv;
10        ICryptoTransform encryptor = aes.CreateEncryptor(aes.Key, aes.IV);
11        using (MemoryStream memoryStream = new MemoryStream())
12        {
13            using (CryptoStream cryptoStream = new CryptoStream((Stream)memoryStream, encryptor,
14                CryptoStreamMode.Write))
15            {
16                using (StreamWriter streamWriter = new StreamWriter((Stream)cryptoStream))
17                {
18                    streamWriter.Write(data);
19                }
20                array = memoryStream.ToArray();
21            }
22        }
23        return Convert.ToBase64String(array);
24    }
```

```

25
26
27 public static string Decrypt(string data, string _key)
28 {
29     string IV = _key.Substring(0, 16);
30     byte[] iv = Encoding.UTF8.GetBytes(IV);
31     byte[] buffer = Convert.FromBase64String(data);
32     using (Aes aes = Aes.Create())
33     {
34         aes.Key = Encoding.UTF8.GetBytes(_key);
35         aes.IV = iv;
36         ICryptoTransform decryptor = aes.CreateDecryptor(aes.Key, aes.IV);
37         using (MemoryStream memoryStream = new MemoryStream(buffer))
38         {
39             using (CryptoStream cryptoStream = new CryptoStream((Stream)memoryStream, decryptor,
CryptoStreamMode.Read))
40             {
41                 using (StreamReader streamReader = new StreamReader((Stream)cryptoStream))
42                 {
43                     return streamReader.ReadToEnd();
44                 }
45             }
46         }
47     }
48 }

```

PHP [↗](#)

```

1 function Encrypt_Decrypt( $action,$msg , $secret_key)
2 {
3     $output = "";
4     $encrypt_method = "AES-256-CBC";
5     $secret_iv =substr($secret_key,0,16);
6     if ( $action == "encrypt" ) {
7         $output = openssl_encrypt($msg, $encrypt_method, $secret_key, OPENSSL_RAW_DATA, $secret_iv);
8         $output = base64_encode($output);
9     }
10    else
11    if( $action == "decrypt" ) {
12        $output = openssl_decrypt(base64_decode($msg), $encrypt_method, $secret_key, OPENSSL_RAW_DATA,
$secret_iv);
13    }
14    return $output;
15 }

```

Examples [↗](#)

Request Body [↗](#)

```

1 {
2     "DeviceInfo": {
3         "PackageName": "com.test.com",
4         "LangCode": "en",
5         "DeviceID": "test_dev_doc"
6     },
7     "Referrer": {
8         "Affiliate": {

```



```

9      "Campaign": "will be shared with you",
10     "ClickID": "your clickid",
11     "Pub_ID": "your pub id",
12     "Aff_ID": "your aff id",
13     "extra": "",
14     "extra1": "",
15     "firstPageButtonID": "msisdn-entry",
16     "secondPageButtonID": "pin-entry",
17     "Country": "the desired country"
18   }
19 },
20 "Request": {
21   "Action": 1,
22   "TransactionID": "b5d7ab80-262e-4246-9dc0-a9ca3202cf74",
23   "SessionID": "",
24   "MSISDN": "",
25   "PinCode": ""
26 }
27 }

```

Responses [🔗](#)

Error [🔗](#)

```

1 {
2   "Error": 1,
3   "Description": "Invalid MSISDN",
4   "MessageToShow": "Please Enter correct MSISDN",
5   "SessionID": "asdad-sdsad-sadsad-sad-asd",
6   "NextAction": {}
7 }

```

Next Action is SendPin, ClicksFlow, ClickToSMS [🔗](#)

```

1 {
2   "Error": 0,
3   "Description": "ok",
4   "MessageToShow": "ok",
5   "SessionID": "asdad-sdsad-sadsad-sad-asd",
6   "NextAction": {
7     "Action": refer to Actions section above,
8     "Name": "Name Of the Action",
9     "Disclaimers": {
10       "headerInfo": "",
11       "middleInfo": "",
12       "footerInfo":
13         "Lorem%20ipsum%20dolor%20sit%20amet%2C%20consectetur%20adipiscing%20elit.%20Morbi%20at%20condimentum%20magna%2C"
14     },
15     "PrivacyPolicy": "https://localhost/PrivacyPolicy",
16     "TermsAndConditions": "https://localhost/TermsAndConditions",
17     "AFScript": "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi at condimentum magna,"
18   }
19 }

```

Next Action is VerifyPin [🔗](#)

```

1 {
2   "Error": 0,

```

```

3      "Description": "ok",
4      "MessageToShow": "ok",
5      "SessionID": "asdad-sdsad-sadsad-sad-asd",
6      "NextAction": {
7          "Action": 3,
8          "Name": "VerifyPin",
9          "Disclaimers": {
10             "headerInfo": "",
11             "middleInfo": "",
12             "footerInfo":
"Lorem%20ipsum%20dolor%20sit%20amet%2C%20consectetur%20adipiscing%20elit.%20Morbi%20at%20condimentum%20magna%2C"
13         },
14         "PinLengths": [4,5],
15         "PrivacyPolicy": "https://localhost/PrivacyPolicy",
16         "TermsAndConditions": "https://localhost/TermsAndConditions",
17         "AFScript": "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi at condimentum magna,"
18     }
19 }

```

Next Action is LoadURL [↗](#)

```

1  {
2      "Error": 0,
3      "Description": "Load URL",
4      "MessageToShow": "",
5      "SessionID": "",
6      "NextAction": {
7          "Action": 4,
8          "Name": "LoadURL",
9          "URL": "http://google.com",
10     }
11 }

```

Next Action is SendSMS [↗](#)

```

1  {
2      "Error": 0,
3      "Description": "",
4      "MessageToShow": "",
5      "SessionID": "",
6      "NextAction": {
7          "Action": 5,
8          "Name": "SendSMS",
9          "Message": "Sub",
10         "Destination": "1234"
11     }
12 }

```

NextAction is Close [↗](#)

```

1  {
2      "Error": 0,
3      "Description": "ok",
4      "MessageToShow": "ok",
5      "SessionID": "",
6      "NextAction": {
7          "Action": 7,
8          "Name": "Close",

```

