etisalat and

CONSENT MANAGEMENT SERVICE

Contents

1. Executive Summary

Below are the new released set of instructions to govern and curb the unsolicited marketing communications in the UAE.

To ensure the new set of instructions are smoothly executed with least possible disruption in the market, Etisalat is launching a new service CMS (Consent Management Service). The service allows all customers to manage:

1. **Consent**

    a. Upload/Modify consent

2. **Sender ID**

    a. Add/Delete sender IDs

3. **Segregation of SMS**

4. **Detailed reports**

The customer of this service can only be brands, which means a corporate entity that owns the brand name or is the primary owner of the A2P communication. Examples of brands are Adidas, Landmark Group, Carrefour, etc. The brand or enterprise may also have connectivity to Etisalat to be able to send the A2P SMS.

1.1 KEY COMPLIANCE REQUIREMENTS

Existing guidelines

- Ensure consent for all promotional SMS IDs secured with brands
- Promotional SMS to be sent only during the permitted time 7:00am – 9:00pm. Every promotional SMS should have an opt-out footer to enable mobile customers to be able to opt-out

Consent management

- Operator to obtain and store all consent
- Operator to verify consent
- Operator to recycle consent for all MSISDNs ceased

Segregation of traffic

- Differentiate between promotional and transactional
- Sub categorisation of sender IDs into segments
- Promotional traffic sender IDs to start with "AD-"
- Allow for blocking all promotional traffic

Spam-compliant SLA

- To respond on all spam complaints within 24 hours

Customers should be ready with following information for easy transition to the new services.

1. Clear and explicit consents should be secured and uploaded in an analogue or digital format as per policy.
2. NOC documents to be ready for re-registration of sender IDs on CMS.
3. For automatic consent, update development to be completed as per shared API.

*Non-compliance with above guidelines can result in the sender ID suspension or financial penalties.

1.2 CONSENT MANAGEMENT SERVICE

This is a new service being launched for our business messaging customers. Using this account, brands/enterprises will be allowed to request sender IDs, which will be unique across the country, and then upload consents against these sender IDs.

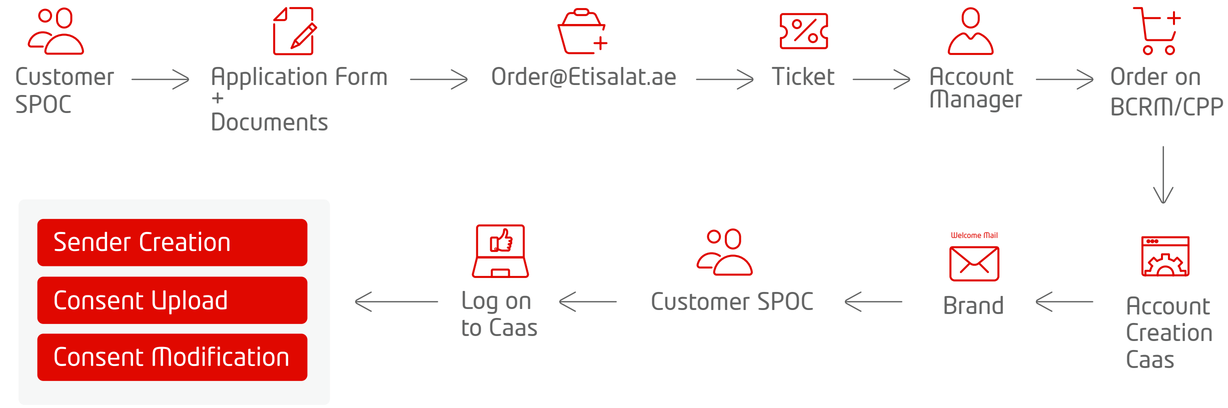
1.2.1 CMS account creation

1. The CMS on-boarding process is completely automated.
2. The account creation can be initiated via the following channels:
   - Sales Manager – ES/Managed SMB/Govt.
   - Channel Partner – Alternate Managed/Managed Indirect
   - Business Centre
3. In case of a new customer account creation, the following customer documents need to be submitted:
   a. Valid Trade license
   b. Passport copy
   c. Power of attorney
   d. Approval letter from the Ministry of Information
   e. Letter from the customer indicating that they are the authorised dealers/owners/ distributors of the brand
4. Customer contact details should be provided in the CMS application form, during the CMS account creation to be able to send email and SMS confirmation via CMS.
   a. Account creation notification
   b. Account update notifications
5. Email with CMS login details will be automatically sent to the customer once the account has been created.
6. The brands will have following options from the CMS portal:
   a. Create and delete sender IDs
   b. Display active sender IDs
   c. Existing consented base
   d. Upload new consents
   e. View uploaded consents

Below is a high-level flow created to depict the automatic CMS account creation process.

1.2.1 CMS account creation

1. The CMS on-boarding process is completely automated.
2. The account creation can be initiated via the following channels:
   - Sales Manager – ES/Managed SMB/Govt.
   - Channel Partner – Alternate Managed/Managed Indirect
   - Business Centre
3. In case of a new customer account creation, the following customer documents need to be submitted:
   a. Valid Trade license
   b. Passport copy
   c. Power of attorney
   d. Approval letter from the Ministry of Information
   e. Letter from the customer indicating that they are the authorised dealers/owners/distributors of the brand
4. Customer contact details should be provided in the CMS application form, during the CMS account creation to be able to send email and SMS confirmation via CMS.
   a. Account creation notification
   b. Account update notifications
5. Email with CMS login details will be automatically sent to the customer once the account has been created.
6. The brands will have following options from the CMS portal:
   a. Create and delete sender IDs
   b. Display active sender IDs
   c. Existing consented base
   d. Upload new consents
   e. View uploaded consents



1.2.2CMS sender ID creation

1. A sender ID should be allowed only once to a single brand across the UAE i.e. no duplication of any sender ID. If a sender ID is on-boarded via Etisalat, then du cannot issue the same sender ID/short code and vice versa.

2. Brands can share the list of required sender IDs with following details:

   a. Promotional:

      i. Compulsory to start with AD-

      ii. 3-8 characters (excluding AD-)

      iii. Cannot be only numeric (excluding AD-)

      iv. Max. 2 special characters allowed (excluding -)

      v. Spaces are allowed, but should not end with space

   b. Transactional:

      i. 3-11 characters

      ii. Can be alpha or alphanumeric or numeric

      iii. Max. 2 special characters allowed

      iv. Spaces are allowed, but should not end with space

3. Sender ID categories and segments are mandatory to create a sender ID.

   Promotional

   Banking services

   Real estate services

   Health services

   Education services

   Retail sale services

   Tourism services

Transactional

Banking services

Real estate services

Health services

Education services

Retail sale services

Tourism services

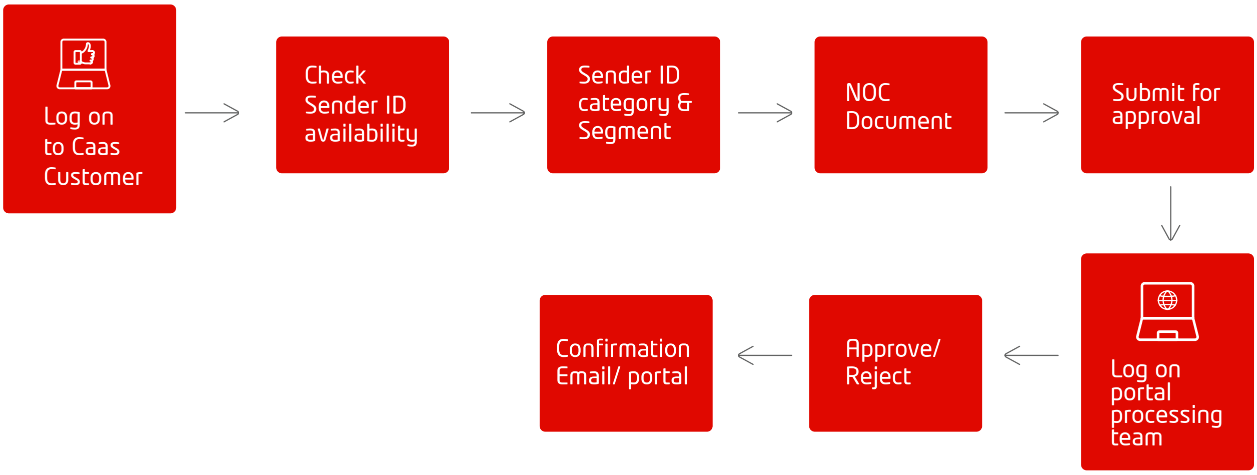
Government

Energy & Utility

a. Sender ID No Objection Certificate (NOC) from the brand/enterprise that owns the sender ID.

4. Sender ID approval or rejected status shall be updated back to the brand via email within 24 - 48 hours.

5. In the approval email to the brand, a Blockchain approval ID for the sender will be shared.


Log on to Caas Customer
Check Sender ID availability
Sender ID category & Segment
NOC Document
Submit for approval
Log on portal processing team
Approve/ Reject
Confirmation Email/ portal

1.3 CONSENT MANAGEMENT PROCESS

1.3.1 Consent guidelines

These fields are the minimum number of fields that are required to be captured by the brand when securing consent.

Analogue consent details

a) MSISDN

b) Date of consent gathering

c) Time of consent gathering (optional)

d) File with customer's signature

Sample File

MSISIDN	DATE-TIME OF CONSENT	FILE NAME
971XXXXXXXXX	YYYY-MM-DD	Promotion-Evidence_971XXXXXXXXX.pdf

Digital consent details

a) MSISDN

b) Date of consent gathering

c) Time of consent gathering (optional)

d) Channel type (web or app)

e) Channel value (URL or app name)

f) Digital ID (system generated ID or log or customer email ID)

Sample File

MSISIDN	DATE-TIME OF CONSENT	CHANNEL	CHANNEL VALUE	Digital ID
971XXXXXXXXX	YYYY-MM-DD HH:MM:SS	Web	https://www.dubai.com	100111222ABC

Rules applicable for each field:

- MSISDN to start with 971
- Channel must be either web or app only
- Channel value: Max. 50 characters
- DIGITAL_ID: Max. 500 characters
- Max CSV file size: 2MB around 50K records
- Evidence file size: 50MB zip file (individual files in PDF or JPG format) for analogue consent

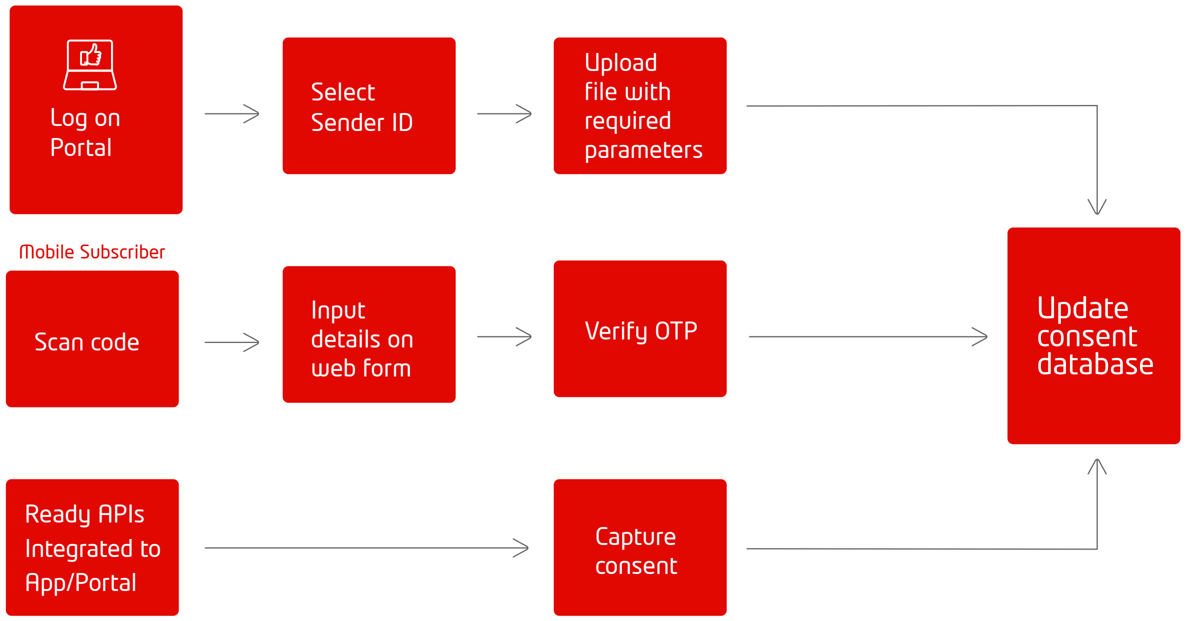
1.3.2 CMS Digital Consent Acquisition

a. Brands will have an option to create a digital consent template to acquire consents via CMS.

b. On template creation, they can download a QR code for each consent template.

c. Brands should be able to print the QR code.

d. Mobile subscribers, on scanning the QR code, must be directed to a portal/ website/app where they need to enter basic details to capture consent



1.4 USAGE SCRUBBING

1. Scrubbing is defined as checking for consent on the CMS before a promotional message is sent to a mobile subscriber. A trigger will be initiated towards the CMS system to verify consent each time a promotional SMS is being sent to a mobile subscriber.

2. All verifications must be done by Etisalat's systems before confirming consent for the SMS to be sent:

   a. Verify if the sender ID is valid.

   b. Verify if the content of the message is as per template or is not spam.

   c. Verify that the customer has provided consent.

   d. Verify that mobile subscriber has not enabled blocking of all promotional sender IDs on his/her account.

3. For transactional message, we will not be checking the availability of consent.

CMS Enterprise Account	Sender IDs	Consents	Scrub Success
Active	Active	Web	Yes
Suspend	Suspend	Active	No
	Disable	Active	No
	Blacklist	Active	No
	Delete	Disable	No

1.5 IMPACT OF MOBILE SUBSCRIBER CESSATION & MNP ON CUSTOMER CONSENTS

1. In case of subscriber cessation and ownership transfer:

   a) All existing customer consents should be inactive and stored for a period of 2 years.

   b) Records of the SMS sent along with consents to be available for a period of 2 years.

2. In case of MNP:

   a) All existing customer consents to be preserved.

   b) Brands must be able to use these consents after MNP to new licensee except telco internal sender ID consents.

1.6 RECOMMENDED GO-LIVE STRATEGY

First, CMS needs to go live with following capabilities:

a. CMS account creation

b. SID creation/deletion

c. Consent upload

After the cut-off planned, after few weeks of the above step.

a. Start consent scrubbing

b. Old legacy SID/short codes to be made inactive

c. After a monitoring period, old SID/SCs to be deleted or cleaned up

1.7 SENDER ID NOC FORMAT

NOC document should be on the company's letterhead.

DATE

Etisalat
Emirates Telecommunications Corporation
Dubai, U.A.E.

TO WHOM IT MAY CONCERN

This is to confirm that we, CLIENT NAME, do not have any objection to register a sender ID with Etisalat UAE.

We undertake to comply with all applicable laws detailed under the UEC Policy (including, but not limited to, as they relate to fraud and spam) and indemnify Etisalat UAE against any action, claim, fine or loss whatsoever incurred because of a breach of law or regulation.

By signing this registration form, we confirm that all SMS will only be sent to eligible customers who have consented (in a form that is recorded and physically presentable) to receive SMS communication in accordance with the details outlined below.

Company name: CLIENT NAME

Type of business/industry: <Business Type>

Sr. No.	Sender ID Type	Company	Sender ID*	SMS Content	Purpose of Use
1	Transactional	<Name>	BrandOTP	Provide Sample	Financial, OTP
2	Promotional	<Name>	AD-BrandPro	Provide Sample	Marketing Comm, Promo

(In case a company is whitelisting sender ID via a reseller (e.g. an aggregator acting on behalf of a bank, retail chain, etc.), the company has to provide an express permission via a signed NOC to the reseller. The reseller needs to attach that signed NOC along with this template for approval).

Thanking you,

....
Signature
Name of Signing Officer
Title
Company Stamp

1.8 CONSENT ACQUISITION API

This REST API section describes the API details to be invoked by a customer to register consents with CMS.

Note: This API is not for bulk consents upload. This is to initiate consent registration request real-time with CMS from a web/mobile app. Bulk upload of consents have to be done from CMS Portal only.

1.8.1 Authorisation token

Invoke this API to get the access and refresh tokens. Access token is required to invoke any further API requests.

The access token is valid for one hour from the time it is created. The user must get a new access token for every one hour.

The refresh token is valid for 30 days.

URL: https://XX.XX.XX.XXX/api/token/

Method: POST

Parameters:

Parameter	Data Type	Description	Remarks
username	**string (40)**	**Username/key configured for the account**	
password	**string (40)**	**Password/secret configured for the account**	

Note: Access token must be passed in authorisation header as bearer token.
Ex: Authorisation: Bearer <access token>

Response:

The response will be in JSON format.

Success response:

```
{

“refresh”:”eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.

eyJ0b2tlbl90eXBlIljoicmVmcmVzaCIsImV4cCI6MTU0ODM5MzU5NCwianRpIjoiYjFlZGFjYzE3ZTcxNGI
wYmIyMzJkYzRlM2ZmYzFiNjMiLCJ1aWQiOjk1fQ.GrUjW4shdeQHk4rKhK19kPs5bedX3baaAt1UgevEtTo”

“access”:”eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIljoiYWNjZXNzIiwiZXhwI
joxNTQ1ODg3OTk0LCJqdGkiOiIwYjVhMmQzMWYyYmM0NmE3YWI2MjZlZDAzMTA4ZDFkNyIsInVpZ
CI6OTV9.joB_MjW1tCrFXGOJGROnIUE0-cBaC5U-mYQ0B6qoqp8”

}
```

Failed response:
If User not found with the credentials:

```
{
   “status”: 401,
   “message”“:Invalid login credentials”
}
```

If account status is not active:

```
{
   “status”:401,
   “message”“: Account Not Active”
}
```

If API access is not enabled:

```
{
   “status”:401,
   “message”“: API access disabled”
}
```

1.8.2 Refresh token

Invoke this API request to get new access token for authorisation.

Refresh token in the response from request (sec 1.1) has to be passed as parameter to this API request.

Refresh token is valid for 30 days; need to get new one using API method in section 1.1

URL: https://XX.XX.XX.XXX/api/token/refresh

Method: POST

Parameters:

Parameter	Data Type	Description	Remarks
refresh	string (40)	refresh token that was received in 1.1	

Response:
The response will be in JSON format.

Success response:

```
{

“access”:”eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwI
joxNTQ1ODg3OTk0LCJqdGkiOiIwYjVhMmQzMWYyYmM0NmE3YWI2MjZlZDAzMTA4ZDFkNyIsInVpZ
CI6OTV9.joB_MjW1tCrFXGOJGROnIUE0-cBaC5U-mYQ0B6qoqp8”

}
```

1.8.3 Register a consent

Invoke this API request to register a consent for a MSISDN.

URL: https://XX.XX.XX.XXX/api/addconsent/

Method: POST

Parameters:

Parameter	Data Type	Description	Remarks
msisdn	String(12)	Valid mobile subscriber MSISDN with country code “971”.	(Mandatory) Ex: 971XXXXXXXXX
template_id	String(20)	An active consent template ID, which is defined in CMS platform.	(Mandatory)
consent_time	String(10)	Consent registration date. Unix timestamp.	(Mandatory) Ex: 1603780067
channel	String(3)	Allowed values: web and app. Case insensitive. E.g.: WEB/Web/ weB also accepted).	(Mandatory)
msisdn channel_val	String(50)	URL address/social platform name or application name.	(Mandatory) minimum 4 characters and maximum 50 characters with all ascii characters allowed.
digital_id	String(500)	System generated ID/log or email address.	(Mandatory) minimum 4 characters and maximum 500 characters with all ascii characters allowed.

Note:

1. This API accepts the request and actual consent registration in CMS systems is asynchronous.
2. Consent is created for each sender ID that is active and attached with the consent template.
3. Consent is ignored for a particular sender ID if there is an active consent already for the same msisdn.

Sample:

Request:

```
{
               “msisdn”: “971XXXXXXXXX”,
                “template_id”: “11XXXXXXXXXXXXXXXXX”,
                “consent_time”: “1603780679”,
                “channel”: “Web”,
                “channel_val”: “facebook.com”,
                “digital_id”: “1234567890”
}
```

Response:

The response will be in JSON format and it contains status, message and a unique request ID(max length 50).

Status 2000 indicates that the request is accepted and request_id is the unique reference number for this transaction.

Success response:

```
{
    “status”: 2000,
    “message”: “Request accepted”,
    “request_id”: “wqd3e432rwer345435436ret53fwshi6”
}
```

Failure responses:

Any status other than 2000 refers to the failure response.

```
{
    “status”: 5020,
    “message”: “Invalid arguments error”,
}
```

1.8.4 Error Codes/Status

Error code(status)	Data Type
401	**Invalid Credentials/Account Not Active**
5000	**Technical error**
5010	**Invalid argument value or type**
5020	**Parameter is missing**
4000	**Bad request**
4040	**No data available. Empty set**

1.8.5 Sample requests

CURL:

```
curl --location --request POST ‘https://XX.XX.XX.XXX/api/addconsent/’ \

--header ‘Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZX
hwIjoxNjEzMTQ1OTQxLCJqdGkiOiI0NDYwYzQ3Nzk4NTI0MTQzOWYxYjJmM2UzMmIzYjAxMSIsInVzZXJfaWQ
iOjEzNX0.OHEbkD1yu0jJ0btn6KSgr-e9SEXTA8kdspeSC8KQ25Y’ \
--header ‘Content-Type: application/json’ \
--data-raw ‘{
   “template_id”: “1108161305948383450”,
   “msisdn”: “971547856897”,
   “channel”: “app”,
   “channel_val”: “89889”,
   “consent_time”: “1612952294”,
   “digital_id”: “8988989”

}’
```

PYTHON:

```
import requests

url = “https://XX.XX.XX.XXX/api/addconsent/”

payload=”{\r\n    \”template_id\”: “1108161305948383450”,\r\n    \”msisdn\”: \”971547856897\”,\r\n    \”chan
nel\”: \”app\”,\r\n    \”channel_val\”: \”89889\”,\r\n    \”consent_time\”: \”1612952294\”,\r\n    \”digital_id\”:
\”8988989\”\r\n}”

headers = {

  ‘Authorization’: ‘Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwI
joxNjEzMTQ1OTQxLCJqdGkiOiI0NDYwYzQ3Nzk4NTI0MTQzOWYxYjJmM2UzMmIzYjAxMSIsInVzZXJfaWQ
iOjEzNX0.OHEbkD1yu0jJ0btn6KSgr-e9SEXTA8kdspeSC8KQ25Y’,

  ‘Content-Type’: ‘application/json’

}

response = requests.request(“POST”, url, headers=headers, data=payload)
```

```
print(response.text)
```

JAVA:

```
OkHttpClient client = new OkHttpClient().newBuilder()
  .build();

MediaType mediaType = MediaType.parse(“application/json”);

RequestBody body = RequestBody.create(mediaType, “{\r\n    \”template_id\”: “1108161305948383450”,\r\n
\”msisdn\”: \”971547856897\”,\r\n    \”channel\”: \”app\”,\r\n    \”channel_val\”: \”89889\”,\r\n    \”consent_
time\”: \”1612952294\”,\r\n    \”digital_id\”: \”8988989\”\r\n}”);
Request request = new Request.Builder()
  .url(“https://XX.XX.XX.XXX/api/addconsent/”)

  .method(“POST”, body)

  .addHeader(“Authorization”, “Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzI
iwiZXhwIjoxNjEzMTQ1OTQxLCJqdGkiOiI0NDYwYzQ3Nzk4NTI0MTQzOWYxYjJmM2UzMmIzYjAxMSIsInVzZXJ
faWQiOjEzNX0.OHEbkD1yu0jJ0btn6KSgr-e9SEXTA8kdspeSC8KQ25Y”)
  .addHeader(“Content-Type”, “application/json”)
  .build();
Response response = client.newCall(request).execute();
```

PHP:

```
<?php
require_once ‘HTTP/Request2.php’;
$request = new HTTP_Request2();
$request->setUrl(‘https://XX.XX.XX.XXX/api/addconsent/’);
$request->setMethod(HTTP_Request2::METHOD_POST);
$request->setConfig(array(

  ‘follow_redirects’ => TRUE

));

$request->setHeader(array(

  ‘Authorization’ => ‘Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXh
wIjoxNjEzMTQ1OTQxLCJqdGkiOiI0NDYwYzQ3Nzk4NTI0MTQzOWYxYjJmM2UzMmIzYjAxMSIsInVzZXJfaWQ
iOjEzNX0.OHEbkD1yu0jJ0btn6KSgr-e9SEXTA8kdspeSC8KQ25Y’,

  ‘Content-Type’ => ‘application/json’
));
$request->setBody(‘{
\n    “template_id”: “1108161305948383450”,
\n    “msisdn”: “971547856897”,
\n    “channel”: “app”,
\n    “channel_val”: “89889”,
\n    “consent_time”: “1612952294”,
\n    “digital_id”: “8988989”
\n}’);
try {
  $response = $request->send();
  if ($response->getStatus() == 200) {
    echo $response->getBody();
  }
  else {
    echo ‘Unexpected HTTP status: ‘ . $response->getStatus() . ‘ ‘ .
    $response->getReasonPhrase();
  }
}
catch(HTTP_Request2_Exception $e) {
  echo ‘Error: ‘ . $e->getMessage();
}
```