Evina OTP Flow Integration Guide

1. Overview

This document describes the integration of Evina's Anti-Fraud solution with OTP-based subscription flows.
It is designed for Mobile Network Operators (MNOs), Payment Gateways, and Digital Merchants who host landing pages and payment pages.

The integration ensures fraud management during the OTP subscription flow by embedding a script on the landing page and verifying subscription requests through Evina's Anti-Fraud platform.

---

2. Fraud Management Flow

1. The merchant requests the **GetScript API** from Evina's Anti-Fraud platform.
2. The merchant installs the returned script on the landing page (or payment page).
3. User events and metrics (e.g., page load, clicks) are sent to Evina's Anti-Fraud platform for analysis.
4. During **Subscription Verification**, the **transactionId (ti)** and **timestamp (ts)** are validated. Based on Evina's analysis, the transaction is either **approved or blocked**.

---

3. GetScript API

The **GetScript API** generates a **Javascript probe** to be inserted into the landing page. This script:

- Is **non-blocking** and dynamically generated via a web service.
- Has **low latency (<100ms RTT included)**.
- Collects real-time signals required for fraud analysis.

Query Parameters

Parameter	Type	Required	Description
ti	string	Yes	Unique transaction identifier. Each transaction must have a unique ti.
te	string	No	Targeted DOM element to protect (CTA button ID). Required in landing page integration. Example: #cta_button.
ts	int32	Yes	Current transaction timestamp (epoch format).
servicename	string	Yes	Service name for transaction identification.
merchantname	string	Yes	Merchant name for transaction identification.
type	string	Yes	Transaction type: "he" or "pin".

Placement

The script must be placed **before all other scripts** inside the <head> tag of the landing page.

---

5. Example: HTML Header with PHP

```
<?php

$html = "http://www.social-sms.com/iq-dcb/dcbprotect.php?action=script&ti=$ti&ts=$ts&te=%23cta_button&servicename=$servicename&merchantname=$merchantname&type=$type";

$test = file_get_contents($html);

$test = json_decode($test);

?>

<head>

 <script>

  <?php echo $test->s; ?>

 </script>

</head>
```

Verification will be handled at the **VMS level** using the ti and ts parameters in subscription or redirection requests.

---

6. OTP Subscription Through Landing Page

Evina supports two OTP subscription page models:

Case I: Single Page (Phone Number + OTP Input on the Same Page)

1. **On page load**, the service provider must:
    - Generate a unique ti (transaction ID) with a service-specific prefix.
    - Generate the current timestamp ts (epoch format).
    - Set merchantname, servicename, and te (e.g., #cta_button).
    - Call **GetScript API** and inject the returned script.

Example CTA button:

<button onclick="functiontodo();" id="cta_button">Subscribe</button>

Example API call:

http://www.social-sms.com/iq-dcb/dcbprotect.php?action=script&ti=SPPref-123456789&ts=1630560419&te=%23cta_button&servicename=testservice&merchantname=spname&type=pin

Inject script:

<script> <?php echo $test->s; ?> </script>

2. User enters **phone number** and clicks **Send Pin Code** → Request sent to SDP.
3. User receives **OTP via SMS** and enters it in the box.
4. User clicks the **Subscribe button (cta_button)**.
5. Service provider sends **VerifyPinCode API** request:

    https://{{ verifyPincode API}}?user={apiuser}&password={apipassword}&msisdn={MSISDN}&shortcode={SHORTCODE}&serviceId={serviceID}&spId={SPID}&pincode={Pincode}&ti=SPPREFIX-123456789&ts=1630560419

6. If response is **success**, user is subscribed.
    If **failure**, reload page with a new ti and ts.

Case II: Two Pages (Phone Number Page + OTP Verification Page)

1. User enters **MSISDN** on Page 1 → Service provider sends OTP request to SDP.
2. User receives **OTP via SMS** and is redirected to OTP verification page.
3. On **OTP verification page load**:
    - Generate ti and ts.
    - Set merchantname, servicename, and te (#cta_button).
    - Call **GetScript API** and inject returned script.

Example button:

<button onclick="functiontodo();" id="cta_button">Subscribe</button>

Example API call:

http://www.social-sms.com/iq-dcb/dcbprotect.php?action=script&ti=SPPref-123456789&ts=1630560419&te=%23cta_button&servicename=testservice&merchantname=spname&type=pin

Inject script:

<script> <?php echo $test->s; ?> </script>

4. User enters OTP → clicks **Subscribe** (cta_button).
5. Service provider sends **VerifyPinCode API** request:
6. https://{{ verifyPincode API }}?user=apiuser&password=apipassword&msisdn=UserMSISDN&
7. shortcode=SHORTCODE&serviceId=serviceID&spId=SPID&pincode=Pincode&
8. ti=SPPREFIX-123456789&ts=1630560419
9. If response is **success**, user is subscribed.
   If **failure**, reload with new ti and ts.