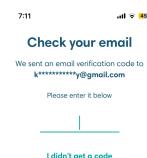# Measure: 2FA

This document showcases a complete breakdown of the two factor authentication journey setup on the sign in journey.

Screen:

2FA Email Funnel Details

The below data is from 30th Sept for the respective markets

	Event Name	Source	% Drop from previous	% From first step	Hypothesis of Drops, and how to identify	Recommendations / Next Steps
1	2FA Exposed	**Front End (Android/iOS)**	US : 100% DE : 100% JP : 100% PT : 100%	100%		

2	Backend 2FA Code Sent ⌄ Definition When backend generates code, this event gets triggered	**Backend** <u>Release:</u> US - 15th Sept JP, IN - 23rd Sept EURBU - 28th Sept	US : 98% DE : 99% JP : 99% PT : 98%	US = 2% DE = 1% JP = 1% PT = 2%	• Investigation happening with developers	
3	Backend Braze Code Sent ⌄ Definition When braze gets notified or updated with a new code to send the email	**Backend** To be developed	-	-	-	-
4	Email Sent	**Braze**	US : 98%	US = 2%	• A measurable gap exists between	ⓘ **Next Steps:**

			DE : 98%	DE = 2%	"Backend 2FA Code Sent" and "Braze Email Sent" timestamps, indicating latency or loss in the handoff to email delivery within the US flow.	Ongoing investigation for newer markets, particularly PT, IN and AT.
			JP : 99%	JP = 1%		
			PT : 97%	PT = 3%		
					• Stakeholders aligned that Braze remains the viable code-sending tool for the US market after review and escalation with the product owner.	
					• Braze support's assessment attributes most failures to user-side issues: invalid or misspelled email addresses, special/wild characters in the email, and full inboxes leading to bounces.	
5	Email Delivered	**Braze**	US : 97%	US = 3%	• Although the email with the 2FA code gets sent to the user, the see "otp limit exceeded" or the "multiple	ℹ️ **RECOMMENDATION:**
			DE : 98%	DE = 2%		• show a countdown on the number of OTP send limit or
			JP : 98%	JP = 2%		

#	Event Name	Source	% Drop from previous	% From first step	Hypothesis of Drops, and how to identify	Recommendations / Next Steps
			PT : 95%	PT = 5%	codes added" error	enforce a cooldown period of 30-60 seconds for the user to request for another OTP • A CAPTCHA can be applied on the last attempt to deter bots
6	2FA Complete	**Front End**	US : 93% DE : 96% JP : 97% PT : 85%	US = 7% DE = 4% JP = 3% PT = 15%	• Users are abandoning during verification, either by exiting mid-flow or after encountering repeated OTP/code errors that halt progression.	

2FA SMS Funnel Details - PL market

#	Event Name	Source	% Drop from previous	% From first step	Hypothesis of Drops, and how to identify	Recommendations / Next Steps
1	2FA Exposed	Frontend	100%			
2	SMS Sent	Twilio	99%	1%		

3	SMS Delivered	Twilio	97.4%	2.6%	**HYPOTHESIS:** • Inbox/Storage full, network issues or incorrect/inactive phone number added	
4	2FA Complete	Frontend	96%	4%	**HYPOTHESIS:** • voluntary exits plus frequent OTP failures are jointly driving drop-offs in the verification step	

How to Identify at a user level WHERE they drop off

```
1   -- User-day 2FA journey table with step timestamps, dropped_off flag, and drop location
2   -- Steps: exposed → backend_2fa_code_sent → email_sent → email_delivered →
    twofa_complete
3
4   DECLARE start_date DATE DEFAULT DATE '2025-09-16';
5   DECLARE end_date   DATE DEFAULT CURRENT_DATE();
6
7   WITH
8   -- 1) Source events (date-pruned, US only)
9   exposed AS (
10    SELECT
11      user_id,
12      country_code,
13      TIMESTAMP_TRUNC(timestamp, SECOND) AS ts,
14      DATE(timestamp) AS twofa_date
15    FROM `dbce-c360-119lake-prod-36f0.f_pampers_hub.v_rew_mobile_track_fct`
16    WHERE event = '2FA Exposed'
17      AND country_code = 'us'
18      AND DATE(timestamp) BETWEEN start_date AND end_date
19  ),
20  backend AS (
21    SELECT
22      user_id,
23      TIMESTAMP_TRUNC(timestamp, SECOND) AS ts,
24    FROM `dbce-c360-119lake-prod-36f0.f_pampers_hub.v_rew_server_track_fct`
25    WHERE event = 'Backend 2FA Code Sent'
26      AND country_code = 'us'
27      AND DATE(timestamp) BETWEEN start_date AND end_date
28  ),
29  email_sent AS (
```

```sql
30    SELECT
31      user_id,
32      TIMESTAMP_TRUNC(braze_timestamp, SECOND) AS ts
33    FROM `dbce-c360-119lake-prod-36f0.f_pampers_hub.v_campaign_response_braze_clt_fct`
34    WHERE braze_event_name = 'Email Sent'
35      AND country_code = 'us'
36      AND DATE(braze_timestamp) BETWEEN start_date AND end_date
37  ),
38  email_delivered AS (
39    SELECT
40      user_id,
41      TIMESTAMP_TRUNC(braze_timestamp, SECOND) AS ts
42    FROM `dbce-c360-119lake-prod-36f0.f_pampers_hub.v_campaign_response_braze_clt_fct`
43    WHERE braze_event_name = 'Email Delivered'
44      AND country_code = 'us'
45      AND DATE(braze_timestamp) BETWEEN start_date AND end_date
46  ),
47  twofa_complete AS (
48    SELECT
49      user_id,
50      TIMESTAMP_TRUNC(timestamp, SECOND) AS ts
51    FROM `dbce-c360-119lake-prod-36f0.f_pampers_hub.v_rew_mobile_track_fct`
52    WHERE event = '2FA Complete'
53      AND country_code = 'us'
54      AND DATE(timestamp) BETWEEN start_date AND end_date
55  ),
56
57  -- 2) First exposure per user/day (anchor for 30m window)
58  first_exposed AS (
59    SELECT
60      user_id,
61      country_code,
62      twofa_date,
63      ts AS exposed_ts
64    FROM (
65      SELECT
66        e.*,
67        ROW_NUMBER() OVER (PARTITION BY user_id, twofa_date ORDER BY ts) AS rn
68      FROM exposed e
69    )
70    WHERE rn = 1
71  ),
72
73  -- 3) Ordered earliest timestamps per step (must follow order; all within 30m of
    exposure)
74  backend_min AS (
75    SELECT fe.user_id, fe.twofa_date, fe.exposed_ts, MIN(b.ts) AS backend_ts
76    FROM first_exposed fe
77    JOIN backend b
78      ON b.user_id = fe.user_id
79     AND b.ts >= fe.exposed_ts
80     AND b.ts <= fe.exposed_ts + INTERVAL 30 MINUTE
81    GROUP BY fe.user_id, fe.twofa_date, fe.exposed_ts
82  ),
83  email_sent_min AS (
84    SELECT fe.user_id, fe.twofa_date, MIN(s.ts) AS email_sent_ts
85    FROM first_exposed fe
86    JOIN backend_min bm
```

```sql
 87       ON bm.user_id = fe.user_id AND bm.twofa_date = fe.twofa_date
 88     JOIN email_sent s
 89       ON s.user_id = fe.user_id
 90      AND s.ts >= bm.backend_ts
 91      AND s.ts <= fe.exposed_ts + INTERVAL 30 MINUTE
 92     GROUP BY fe.user_id, fe.twofa_date
 93  ),
 94  email_delivered_min AS (
 95    SELECT fe.user_id, fe.twofa_date, MIN(d.ts) AS email_delivered_ts
 96    FROM first_exposed fe
 97    JOIN email_sent_min es
 98      ON es.user_id = fe.user_id AND es.twofa_date = fe.twofa_date
 99    JOIN email_delivered d
100      ON d.user_id = fe.user_id
101     AND d.ts >= es.email_sent_ts
102     AND d.ts <= fe.exposed_ts + INTERVAL 30 MINUTE
103    GROUP BY fe.user_id, fe.twofa_date
104  ),
105  complete_min AS (
106    SELECT fe.user_id, fe.twofa_date, MIN(c.ts) AS complete_ts
107    FROM first_exposed fe
108    JOIN email_delivered_min ed
109      ON ed.user_id = fe.user_id AND ed.twofa_date = fe.twofa_date
110    JOIN twofa_complete c
111      ON c.user_id = fe.user_id
112     AND c.ts >= ed.email_delivered_ts
113     AND c.ts <= fe.exposed_ts + INTERVAL 30 MINUTE
114    GROUP BY fe.user_id, fe.twofa_date
115  )
116
117  -- 4) Final user-day table with requested column order
118  SELECT
119    fe.user_id,
120    IF(cm.complete_ts IS NULL, TRUE, FALSE) AS dropped_off,                       -
     - 2nd column
121    CASE                                                                      --
     3rd column
122      WHEN cm.complete_ts IS NOT NULL THEN 'none'
123      WHEN ed.email_delivered_ts IS NOT NULL THEN 'after_email_delivered'
124      WHEN es.email_sent_ts IS NOT NULL THEN 'after_email_sent'
125      WHEN bm.backend_ts IS NOT NULL THEN 'after_backend_2fa_code_sent'
126      ELSE 'after_exposed'
127    END AS dropped_off_location,
128    fe.country_code,
129    fe.twofa_date,
130    fe.exposed_ts                          AS exposed_ts,
131    bm.backend_ts                          AS backend_2fa_code_sent_ts,
132    es.email_sent_ts                       AS email_sent_ts,
133    ed.email_delivered_ts                  AS email_delivered_ts,
134    cm.complete_ts                         AS twofa_complete_ts
135  FROM first_exposed fe
136  LEFT JOIN backend_min          bm ON bm.user_id = fe.user_id AND bm.twofa_date =
     fe.twofa_date
137  LEFT JOIN email_sent_min       es ON es.user_id = fe.user_id AND es.twofa_date =
     fe.twofa_date
138  LEFT JOIN email_delivered_min ed ON ed.user_id = fe.user_id AND ed.twofa_date =
     fe.twofa_date
```

```
139  LEFT JOIN complete_min        cm ON cm.user_id = fe.user_id AND cm.twofa_date =
     fe.twofa_date
140  where fe.user_id is not null and length(fe.user_id) > 5
141  ORDER BY fe.twofa_date DESC, fe.user_id;
142
```