

Quick Reference: Click Tracking & S2S Postback

Key Concepts

1. Click ID

- **What:** Unique UUID v4 identifier for each user click
- **Where:** Generated by backend when user clicks "Start Task"
- **Format:** 550e8400-e29b-41d4-a716-446655440000
- **Purpose:** Links user click to conversion later

2. Tracking Link (`redirectUrl`)

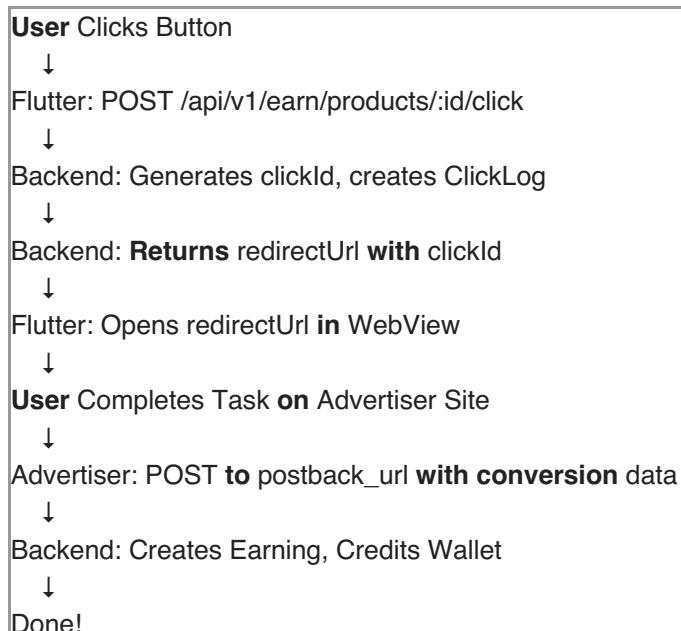
- **What:** The URL user visits to complete the task
- **Contains:** Original `taskUrl` + `click_id` + `postback_url` parameters
- **Example:**

```
https://advertiser.com/offer?click_id=abc-123&postback_url=https://yourapp.com/api/v1/earn/postback?click_id=abc-123
```

3. Postback URL (S2S)

- **What:** Endpoint that advertiser calls when user completes task
- **Format:** `https://yourapp.com/api/v1/earn/postback?click_id={clickId}`
- **Called by:** Advertiser's server (not your app)
- **Purpose:** Notify your backend that conversion happened

Flow Summary



Flutter Integration (3 Steps)

Step 1: Call Generate Click API

```
POST /api/v1/earn/products/{productId}/click
Headers: Authorization: Bearer {token}
Body: { "taskUrl": "https://advertiser.com/offer" }

Response:
{
  "clickId": "abc-123...",
  "redirectUrl": "https://advertiser.com/offer?click_id=abc-123&postback_url=...",
  "expiresAt": "2024-01-02T12:00:00Z"
}
```

Step 2: Open redirectUrl in WebView

```
Navigator.push(
  context,
  MaterialPageRoute(
    builder: (context) => WebViewScreen(url: redirectUrl),
  ),
);
```

Step 3: Done!

The advertiser will automatically call the postback URL when conversion happens. Your backend handles everything else.

API Endpoints

Endpoint	Method	Auth	Purpose
/api/v1/earn/products/:id/click	POST	Required	Generate click, get tracking URL
/api/v1/earn/track/:clickId	GET	Optional	Analytics tracking (optional)
/api/v1/earn/postback?click_id=...	POST	Public	Called by advertiser (S2S)

Flutter Code Template

```

// 1. Generate click
final api = EarningsApi(token: userToken);
final response = await api.generateClick(
  productId: product.id,
  taskUrl: product.taskUrl,
);

// 2. Open in WebView
Navigator.push(
  context,
  MaterialPageRoute(
    builder: (context) => TaskWebViewScreen(
      redirectUrl: response.redirectUrl,
      clickId: response.clickId,
    ),
  ),
);

// 3. That's it! Postback happens automatically

```

Where Things Are Stored

Data	Location	Purpose
Click ID	Response from generateClick	Track this click
Tracking URL	Response from generateClick	Open this in WebView
Click Log	MongoDB ClickLog collection	Backend tracking
Earning	MongoDB Earning collection	User credit record
Postback URL	Automatically included in redirectUrl	Advertiser calls this

⚠ Important Notes

- Always use redirectUrl** - Never send user directly to taskUrl
- Click expires in 24 hours** - User must complete task within 24h
- Postback is automatic** - You don't need to do anything after opening WebView
- One click = One conversion** - System prevents duplicate credits
- Wallet credit happens automatically** - When postback received with status="completed"

Test Flow

1. Test Click Generation

```

curl -X POST http://localhost:3000/api/v1/earn/products/PRODUCT_ID/click \
-H "Authorization: Bearer TOKEN" \
-H "Content-Type: application/json" \
-d '{"taskUrl": "https://example.com"}'

```

2. Test Postback (Simulate Advertiser)

```
curl -X POST "http://localhost:3000/api/v1/earn/postback?click_id=CLICK_ID" \
-H "Content-Type: application/json" \
-d '{
  "amount": 50.00,
  "status": "completed",
  "transactionId": "test_123"
}'
```

Complete Example

See FLUTTER_INTEGRATION_EXAMPLE.dart for full working code!

Related Files

- **Full Documentation:** CLICK_TRACKING_S2S_DOCUMENTATION.md
- **Flutter Code:** FLUTTER_INTEGRATION_EXAMPLE.dart
- **Backend Controller:** src/controllers/earnings.controller.ts
- **Models:** src/models/ClickLog.ts, src/models/Earning.ts