Earnings System Documentation

Table of Contents

Overview

The Earnings System is a comprehensive solution for tracking user earnings from product applications, offer completions, and referral commissions. It integrates with external advertisers/networks through Server-to-Server (S2S) postback tracking and includes a built-in affiliate/referral program.

Key Features

- ✅ **Product Application Tracking**: Users can apply for products and track their applications
- ✅ **S2S Postback Tracking**: Automatic earning creation when offers are completed
- ✅ **Wallet Management**: Automatic crediting of earnings to user wallets
- ✅ **Referral Commissions**: Automatic commission calculation and crediting for referrals
- ✅ **Earning History**: Complete history of all earnings with filtering and pagination
- ✅ **Status Management**: Track earnings from pending to completed

---

System Architecture

Core Components

```
┌──────────────┐
│    User      │
│  (Referrer)  │
└──────┬───────┘
       │
       │ Refers
       ▼
┌──────────────┐     ┌──────────────┐     ┌──────────────┐
│    User      │────▶│ Application  │────▶│   Product    │
│  (Referred)  │     │              │     │              │
└──────┬───────┘     └──────┬───────┘     └──────────────┘
       │                    │
       │                    │ Completes Offer
       │                    ▼
       │             ┌──────────────┐
       │             │   Postback   │
       │             │    (S2S)     │
```

```
      │              └──────┬───────┘
      │                     │
      │                     ▼
      │              ┌─────────────┐
      │              │  Earning    │
      │              │  (User B)   │
      │              └──────┬───────┘
      │                     │
      │                     │ 10% Commission
      │                     ▼
      │              ┌─────────────┐
      │              │  Earning    │
      │              │  (User A)   │
      │              │ Commission  │
      │              └─────────────┘
      │
      │
      └─────────────────────────────┐
                                    │
                                    ▼
                           ┌─────────────┐
                           │   Wallet    │
                           │  (Credited) │
                           └─────────────┘
```

Earning Lifecycle

Step 1: User Applies for Product

Endpoint: `POST /api/v1/earn/products/:productId/apply`

What Happens:

1. User submits application with client details and documents
2. System creates an `Application` record with status `pending`
3. System generates a unique `trackingToken` (64-character hex string)
4. System returns `postbackUrl` for the advertiser to configure

Request Example:

```
POST /api/v1/earn/products/507f1f77bcf86cd799439011/apply
Authorization: Bearer <token>

{
  "clientDetails": {
    "clientName": "John Doe",
    "businessName": "Doe Enterprises",
    "gstin": "29ABCDE1234F1Z5",
    "addressProof": "https://example.com/proof.pdf"
  },
  "documents": {
    "aadhar": "https://example.com/aadhar.pdf",
    "pan": "https://example.com/pan.pdf"
  },
  "offerId": "507f1f77bcf86cd799439012"
}
```

Response Example:

```
{
  "success": true,
  "message": "Application submitted successfully",
  "data": {
    "applicationId": "507f1f77bcf86cd799439013",
    "productId": "507f1f77bcf86cd799439011",
    "status": "pending",
    "submittedAt": "2024-01-15T10:30:00.000Z",
    "trackingToken":
"a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0u1v2w3x4y5z6a7b8c9d0e1f2",
    "postbackUrl": "https://yourapp.com/api/v1/earn/postback?
token=a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0u1v2w3x4y5z6a7b8c9d0e1f2"
  }
}
```

Step 2: Advertiser Configures Postback URL

The advertiser/network receives the `postbackUrl` and configures it in their system. When the offer is completed, they will send a POST request to this URL.

Step 3: Offer Completion - Postback Received

Endpoint: `POST /api/v1/earn/postback?token={trackingToken}`

What Happens:

1. Advertiser sends POST request with completion data
2. System finds application by `trackingToken`
3. System creates or updates `Earning` record
4. If status is `completed` :
   - Credits user's wallet
   - Updates application status to `approved`
   - Processes referral commission (if applicable)
5. Returns success response

Postback Request Example:

```
POST /api/v1/earn/postback?
token=a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0u1v2w3x4y5z6a7b8c9d0e1f2

{
  "amount": 500,
  "status": "completed",
  "offerId": "507f1f77bcf86cd799439012",
  "transactionId": "TXN123456789",
  "conversionId": "CONV987654321"
}
```

Postback Response:

```
{
  "success": true,
  "message": "Postback received and processed",
  "data": {
    "earningId": "507f1f77bcf86cd799439014",
    "applicationId": "507f1f77bcf86cd799439013",
    "status": "completed",
    "amount": 500
  }
}
```

Step 4: Earning Status Flow

```
Application Created
       │
       ▼
Earning Created (status: pending)
       │
       ▼
Postback Received (status: completed)
       │
       ├──→ Wallet Credited
       │
       ├──→ Application Status: approved
       │
       └──→ Referral Commission Processed (if applicable)
```

Postback Tracking (S2S)

How It Works

1. **Unique Tracking Token**: Each application gets a unique 64-character hexadecimal token
2. **Postback URL Generation**: System generates a URL with the token as query parameter
3. **Advertiser Configuration**: Advertiser configures this URL in their system
4. **Automatic Tracking**: When offer completes, advertiser sends POST request
5. **Earning Creation**: System automatically creates/updates earning record

Postback Data Structure

The postback endpoint accepts:

- **Query Parameter**: `token` (required) - The tracking token
- **Body Parameters**:
  - `amount` (optional) - Earning amount (uses product's `earnUpTo` if not provided)
  - `status` (optional) - Status: `pending`, `completed`, `approved`, `rejected`, `cancelled`

  - `offerId` (optional) - ID of the offer that was completed
  - `transactionId` (optional) - Transaction ID from advertiser
  - `conversionId` (optional) - Conversion ID from advertiser
  - Any additional data is stored in `postbackData` field

Security

- **No Authentication Required**: Postback endpoint is public (secured by unique token)
- **Token Validation**: System validates token exists and finds corresponding application
- **Idempotency**: System prevents double-crediting by checking existing earnings
- **Error Handling**: Returns 200 OK even on errors to prevent postback retries

---

Affiliate/Referral System

Overview

When a user refers another user, and the referred user completes an offer, the referrer automatically earns a commission (default 10%).

Registration with Referral

Endpoint: `POST /api/v1/auth/register`

Request with Referral Code:

```
{
  "name": "Jane Doe",
  "email": "jane@example.com",
  "phone": "9876543210",
  "password": "password123",
  "confirmPassword": "password123",
  "agreeToTerms": true,
  "referralCode": "ABC12345"  // Optional referral code
}
```

What Happens:

1. System validates referral code exists
2. System prevents self-referral
3. Creates user with `referredBy` field set
4. Creates `Referral` record linking referrer and referred user
5. Updates referrer's `totalReferrals` count

Commission Calculation

When a referred user completes an offer:

1. **Original Earning**: User B earns ₹500
2. **Commission Calculation**: ₹500 × 10% = ₹50
3. **Commission Earning**: System creates earning for User A (referrer)
4. **Automatic Crediting**: ₹50 credited to User A's wallet
5. **Stats Update**: Referral record and user stats updated

Commission Flow

```
User A (Referrer)
    │
    │ Shares Code: ABC12345
    │
    ▼
User B (Referred) Registers
    │
    │ Applies for Product
    │
    │ Completes Offer → Earns ₹500
    │
    ▼
Postback Received
    │
    ├───▶ User B Earning Created (₹500)
    │     └───▶ Wallet Credited: ₹500
    │
    └───▶ Commission Calculated (10% = ₹50)
          └───▶ User A Earning Created (₹50)
                └───▶ Wallet Credited: ₹50
```

Referral Dashboard

Endpoint: `GET /api/v1/referral/dashboard`

Response Example:

```
{
  "success": true,
  "data": {
    "referralCode": "ABC12345",
    "referralLink": "https://yourapp.com/register?ref=ABC12345",
    "stats": {
      "totalReferrals": 10,
      "activeReferrals": 7,
      "totalCommissions": 5000,
      "thisMonthCommissions": 1500,
      "referralEarnings": 5000,
      "commissionRate": 10
    },
    "referrals": [
      {
        "id": "507f1f77bcf86cd799439015",
        "referredUser": {
          "id": "507f1f77bcf86cd799439016",
          "name": "Jane Doe",
          "email": "jane@example.com",
          "phone": "9876543210",
          "joinedAt": "2024-01-10T08:00:00.000Z"
        },
        "status": "active",
        "totalCommissions": 500,
        "lastCommissionAt": "2024-01-15T10:30:00.000Z",
        "createdAt": "2024-01-10T08:00:00.000Z"
      }
    ]
  }
}
```

Data Models

Earning Model

```
{
  userId: ObjectId,              // User who earned
  productId: ObjectId,          // Product that generated earning
  applicationId: ObjectId,       // Application that led to earning
  offerId: ObjectId,            // Optional: Specific offer completed
  amount: Number,               // Earning amount
  status: String,               // 'pending' | 'completed' | 'cancelled'
  type: String,                 // 'offer_completion' | 'referral_commission'
  earnedAt: Date,               // When earning was recorded
  creditedAt: Date,             // When wallet was credited
  postbackData: Object,         // Store postback data for reference

  // Referral commission fields
  isReferralCommission: Boolean, // True if this is a commission
  referrerId: ObjectId,         // User who earned commission
  referredUserId: ObjectId,      // User whose earning generated commission
  referralCommissionRate: Number // Commission rate used (e.g., 0.1 for 10%)
}
```

Application Model

```
{
  userId: ObjectId,             // User who applied
  productId: ObjectId,          // Product applied for
  clientDetails: Object,        // Client information
  documents: Object,            // Required documents
  status: String,              // 'pending' | 'in_review' | 'approved' | 'rejected'
  trackingToken: String,        // Unique token for postback tracking
  timeline: Array,             // Status timeline
  submittedAt: Date
}
```

Referral Model

```
{
  referrerId: ObjectId,         // User who made referral
  referredUserId: ObjectId,     // User who was referred (unique)
  referralCode: String,         // Code that was used
  status: String,               // 'pending' | 'active' | 'inactive'
  totalCommissions: Number,     // Total commissions earned
  lastCommissionAt: Date        // Last commission timestamp
}
```

User Model (Relevant Fields)

```
{
  referralCode: String,         // Unique 8-character code
  referredBy: ObjectId,         // User who referred this user
  referralEarnings: Number,     // Total earnings from referrals
  totalReferrals: Number,       // Total users referred
  activeReferrals: Number,      // Active referrals (who have earned)
  walletBalance: Number,        // Current wallet balance
  totalEarnings: Number         // Total lifetime earnings
}
```

API Endpoints

Earnings Endpoints

Get Earnings Dashboard

```
GET /api/v1/earn/dashboard?section=sambhav&category=campaign
```

Returns wallet balance, total earnings, leads, sales, and products.

Get Earnings Products

```
GET /api/v1/earn/products?section=sambhav&category=campaign&page=1&limit=20
```

Returns paginated list of earning products.

Get Product Offers

```
GET /api/v1/earn/products/:productId/offers
```

Returns active offers for a product.

Get Product Detail

```
GET /api/v1/earn/products/:productId/detail
```

Returns product details with user's metrics (total earnings, leads, sales).

Apply for Product

```
POST /api/v1/earn/products/:productId/apply
```

Creates application and returns tracking token and postback URL.

Get Application Status

```
GET /api/v1/earn/applications/:applicationId/status
```

Returns application status and timeline.

Get Earnings History

```
GET /api/v1/earn/earnings?page=1&limit=20&startDate=2024-01-01&endDate=2024-01-31
```

Returns paginated earnings history with summary stats.

Postback Endpoint (S2S)

```
POST /api/v1/earn/postback?token={trackingToken}
```

Receives postback from advertiser when offer is completed.

Withdraw Earnings

```
POST /api/v1/earn/withdraw
```

Creates withdrawal request.

Get Withdrawals

```
GET /api/v1/earn/withdrawals?page=1&limit=20&status=pending
```

Returns withdrawal history.

Referral Endpoints

Get Referral Dashboard

```
GET /api/v1/referral/dashboard
```

Returns referral code, link, stats, and referral list.

Get Referrals

```
GET /api/v1/referral/referrals?page=1&limit=20&status=active
```

Returns paginated list of referrals with commission details.

Get Referral Commissions

```
GET /api/v1/referral/commissions?page=1&limit=20&startDate=2024-01-01&endDate=2024-01-31
```

Returns commission earnings history.

Verify Referral Code

```
GET /api/v1/referral/verify-code?code=ABC12345
```

Verifies if a referral code is valid.

Workflow Examples

Example 1: Complete Earning Flow

1. **User Applies for Product**

   ```
   POST /api/v1/earn/products/123/apply
   → Application created
   → trackingToken: "abc123..."
   → postbackUrl: "https://app.com/api/v1/earn/postback?token=abc123..."
   ```

2. **Advertiser Configures Postback**

   - Advertiser receives postbackUrl
   - Configures in their system

3. **User Completes Offer**

   - User completes offer on advertiser's platform
   - Advertiser sends postback:

   ```
   POST /api/v1/earn/postback?token=abc123...
   Body: { "amount": 500, "status": "completed" }
   ```

4. **System Processes**

   - Creates Earning record (₹500, status: completed)
   - Credits user's wallet: ₹500
   - Updates application status: approved
   - If user was referred, processes commission

5. **User Views Earnings**

   ```
   GET /api/v1/earn/earnings
   → Shows earning of ₹500
   → Wallet balance updated
   ```

Example 2: Referral Commission Flow

1. **User A Shares Referral Code**

   - Code: `ABC12345`

    - Link: `https://app.com/register?ref=ABC12345`

2. **User B Registers with Code**

   ```
   POST /api/v1/auth/register
   Body: { ..., "referralCode": "ABC12345" }
   → User B created with referredBy: User A
   → Referral record created
   → User A's totalReferrals: +1
   ```

3. **User B Completes Offer**

    - User B earns ₹1000
    - Postback received and processed

4. **Commission Automatically Processed**

    - Commission: ₹1000 × 10% = ₹100
    - User A's commission earning created (₹100)
    - User A's wallet credited: ₹100
    - User A's referralEarnings: +₹100
    - Referral record updated (status: active, totalCommissions: ₹100)

5. **Both Users View Earnings**

    - User B sees: ₹1000 earning
    - User A sees: ₹100 commission earning

Configuration

Environment Variables

Add to `.env` file:

```
# Base URL for generating postback URLs
BASE_URL=https://yourapp.com

# Referral commission rate (0.1 = 10%, 0.15 = 15%, etc.)
REFERRAL_COMMISSION_RATE=0.1
```

Commission Rate

The commission rate is configurable via `REFERRAL_COMMISSION_RATE`:

- Default: `0.1` (10%)
- Example: `0.15` = 15% commission
- Example: `0.2` = 20% commission

Key Features & Benefits

1. Automatic Tracking

- No manual intervention needed

- Postback system handles everything automatically
- Real-time wallet crediting

2. **Double-Credit Prevention**

- System checks for existing earnings before crediting
- Idempotent postback processing
- Safe to retry postbacks

3. **Complete Audit Trail**

- All postback data stored in `postbackData` field
- Timeline tracking in applications
- Complete earning history

4. **Referral Automation**

- Automatic commission calculation
- Automatic wallet crediting
- Real-time stats updates

5. **Scalable Architecture**

- Indexed database queries
- Efficient aggregation for stats
- Pagination for large datasets

Error Handling

Postback Errors

- Always returns 200 OK to prevent retries
- Errors logged but don't break flow
- Invalid tokens return 200 with error message

Referral Errors

- Invalid referral codes return 400 error
- Self-referral prevented
- Duplicate referrals prevented (unique constraint)

Earning Errors

- Missing applications return 404
- Invalid amounts default to product's `earnUpTo`
- Status validation ensures only valid statuses

Best Practices

1. **Postback URL Security**

    - Keep tracking tokens secret
    - Use HTTPS for postback URLs
    - Monitor postback logs for suspicious activity

2. **Commission Rate**

  - Set reasonable commission rates
  - Consider business model and margins
  - Can be adjusted per product if needed (future enhancement)

3. **User Experience**
   - Show clear earning status
   - Provide timeline for applications
   - Display referral stats prominently

4. **Monitoring**
   - Monitor postback success rates
   - Track commission payouts
   - Alert on unusual patterns

Support & Troubleshooting

Common Issues

1. **Postback Not Received**
   - Verify postback URL is correctly configured
   - Check tracking token is correct
   - Ensure advertiser system is sending POST requests

2. **Commission Not Credited**
   - Verify referral relationship exists
   - Check commission rate configuration
   - Ensure original earning was completed

3. **Duplicate Earnings**
   - System prevents this automatically
   - Check for duplicate postback calls
   - Verify idempotency logic

Summary

The Earnings System provides a complete solution for:

- ✅ Tracking product applications
- ✅ Automatic earning creation via S2S postback
- ✅ Wallet management and crediting
- ✅ Referral commission automation
- ✅ Complete earning history and analytics
- ✅ Scalable and secure architecture

All processes are automated, secure, and designed to scale with your business needs.