Admin Credit Management Guide

Overview

Your Roblox tool already has a complete admin credit management system! This guide explains how to use it to manually add credits to customer accounts (perfect for demo accounts or promotional credits).

✅ What Already Exists

1. Admin API Endpoint (`/api/admin/credits/add`)

- **Location**: `/src/app/api/admin/credits/add/route.ts`
- **Method**: POST
- **Authentication**: Requires SUPER_ADMIN or CUSTOMER_ADMIN role
- **Features**:
- Validates all inputs (amount, customer ID, description)
- Enforces maximum limit of 10,000 credits per transaction
- Records transactions with special admin payment IDs
- Logs admin actions with username for audit purposes
- Returns detailed transaction information

2. Database Schema

Your database already has all necessary tables:
- `customer_credits` : Stores credit balances for each customer
- `credit_transactions` : Records all credit additions/usage with full audit trail
- `customers` : Customer information
- Transaction constraints ensure data integrity

🆕 What We've Added

1. Enhanced Customer Management UI

We've updated the admin dashboard's Customer Management component to include:

Credit Display

- **Credit Balance**: Shows current credits for each customer
- **Total Purchased**: Lifetime credits purchased
- **Total Used**: Lifetime credits consumed
- All displayed in the Customers table for easy viewing

Add Credits Button

- Green "💳 Add Credits" button for each customer
- Opens a modal dialog for easy credit addition
- Shows current balance before adding
- Input validation and helpful hints

Add Credits Modal Features

- Customer name and current balance display
- Amount input (1-10,000 credits)
- Description field for audit trail
- Warning about admin credit addition
- Success confirmation with new balance
- Error handling with detailed messages

2. Updated Database View

We've enhanced the `customer_stats` view to include:
- `credit_balance` : Current credit balance
- `total_credits_purchased` : Total credits purchased
- `total_credits_used` : Total credits used
- Plus previously missing fields: `contact_email` , `max_users` , `logo_url`

3. Migration Script

- **File**: `/database/migrations/013_add_credits_to_customer_stats.sql`
- **Purpose**: Updates the database view to show credit information
- **Documentation**: `/database/migrations/APPLY_MIGRATION_013.md`

🚀 How to Deploy

Step 1: Run the Database Migration

You have three options:

Option A: Using Node.js Script (Recommended)

```
cd /home/ubuntu/roblox-tool
node scripts/run-migration-013.js
```

Option B: Direct psql

```
psql -U your_username -d your_database -f database/migrations/
013_add_credits_to_customer_stats.sql
```

Option C: Copy-Paste in psql

```
# Connect to your database
psql -U your_username -d your_database

# Then paste the SQL from the migration file
```

Step 2: Verify the Migration

```sql
-- Check if the view has the new columns
SELECT column_name, data_type
FROM information_schema.columns
WHERE table_name = 'customer_stats'
AND column_name IN ('credit_balance', 'total_credits_purchased', 'total_credits_used')
;

-- Test the view
SELECT id, name, credit_balance, total_credits_purchased, total_credits_used
FROM customer_stats
LIMIT 5;
```

Step 3: Restart Your Application (if needed)

```bash
# If using PM2
pm2 restart roblox-tool

# If using Next.js dev server
# Just stop and restart: npm run dev

# If using Docker
docker-compose restart
```

📖 How to Use

Adding Credits to a Customer Account

1. **Log in** as a super admin to your application
   - URL: `https://your-domain.com/admin` (or `/admin` on your deployment)

2. **Navigate to the Customers tab**
   - You'll see the admin dashboard
   - Click on the "Customers" tab

3. **View Customer Information**
   - You'll see a table with all customers
   - Each row shows:

     - Customer name and ID
     - Status (Active/Inactive)
     - User count
     - **Credit balance** (new!)
     - Total searches
     - Created date
     - Actions

4. **Click "💳 Add Credits"**
   - Find the customer you want to add credits to
   - Click the green "💳 Add Credits" button in the Actions column

5. **Fill in the Modal**
   - **Amount**: Enter the number of credits (1-10,000)

- **Description**: Enter a reason (e.g., "Demo account credits", "Promotional credits for Q4 campaign")
- Review the current balance shown at the top

6. **Submit**
   - Click "Add Credits"
   - You'll see a success message with the new balance
   - The customer table will refresh automatically

Example Use Cases

Demo Account

```
Amount: 100
Description: Demo account credits for prospect evaluation
```

Promotional Credits

```
Amount: 500
Description: Q1 2024 promotional credits - New customer bonus
```

Compensation

```
Amount: 50
Description: Service interruption compensation - Incident #1234
```

Testing

```
Amount: 1000
Description: Internal testing credits - QA Team
```

🔍 Audit Trail

Every credit addition is recorded in the `credit_transactions` table with:
- Transaction ID
- Customer ID
- User ID (who added the credits)
- Amount added
- Balance before and after
- Description (includes admin username)
- Payment ID (format: `MANUAL_ADMIN_{userId}_{timestamp}`)
- Timestamp

Viewing Transaction History

Query the database to see all admin credit additions:

```sql
-- All admin credit additions
SELECT
    ct.id,
    ct.created_at,
    c.name as customer_name,
    u.username as admin_user,
    ct.amount,
    ct.balance_before,
    ct.balance_after,
    ct.description
FROM credit_transactions ct
JOIN customers c ON ct.customer_id = c.id
JOIN users u ON ct.user_id = u.id
WHERE ct.payment_id LIKE 'MANUAL_ADMIN_%'
ORDER BY ct.created_at DESC;
```

```sql
-- Admin credits for a specific customer
SELECT
    ct.created_at,
    u.username as added_by,
    ct.amount,
    ct.description
FROM credit_transactions ct
JOIN users u ON ct.user_id = u.id
WHERE ct.customer_id = YOUR_CUSTOMER_ID
AND ct.payment_id LIKE 'MANUAL_ADMIN_%'
ORDER BY ct.created_at DESC;
```

🔒 Security Features

Access Control

- **SUPER_ADMIN**: Can add credits to any customer
- **CUSTOMER_ADMIN**: Can only add credits to their own customer account
- All other users: No access (403 Forbidden)

Validation

- Amount must be positive
- Amount cannot exceed 10,000 per transaction
- Description is required
- Customer must exist

Audit Logging

- Every action is logged with admin username
- Transaction ID for tracking
- Timestamps for all actions
- Cannot be deleted (data integrity)

⚙️ API Usage (Alternative Method)

If you prefer to use the API directly (e.g., from scripts or other tools):

Endpoint

```
POST /api/admin/credits/add
```

Headers

```
Content-Type: application/json
Cookie: next-auth.session-token=YOUR_SESSION_TOKEN
```

Request Body

```json
{
  "customerId": 1,
  "amount": 100,
  "description": "Demo account credits"
}
```

Response (Success)

```json
{
  "success": true,
  "message": "Credits added successfully",
  "transaction": {
    "id": 123,
    "customerId": 1,
    "amount": 100,
    "balanceBefore": 0,
    "balanceAfter": 100,
    "description": "[ADMIN] Demo account credits (Added by: admin)",
    "createdAt": "2024-12-01T10:30:00.000Z"
  }
}
```

Response (Error)

```json
{
  "error": "Forbidden - Admin access required",
  "details": "You must be a SUPER_ADMIN or CUSTOMER_ADMIN to add credits"
}
```

Example using curl

```bash
curl -X POST https://your-domain.com/api/admin/credits/add \
  -H "Content-Type: application/json" \
  -H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN" \
  -d '{
    "customerId": 1,
    "amount": 100,
    "description": "Demo account credits"
  }'
```

Example using Postman/Insomnia

1. Create a new POST request
2. URL: `https://your-domain.com/api/admin/credits/add`
3. Headers: `Content-Type: application/json`
4. Body (JSON):
   ```
   json
     {
       "customerId": 1,
       "amount": 100,
       "description": "Demo account credits"
     }
   ```
5. Make sure you're logged in (cookies will be sent automatically)

📊 Monitoring

Check Customer Credit Balances

```sql
SELECT
    c.id,
    c.name,
    COALESCE(cc.balance, 0) as balance,
    COALESCE(cc.total_purchased, 0) as total_purchased,
    COALESCE(cc.total_used, 0) as total_used
FROM customers c
LEFT JOIN customer_credits cc ON c.id = cc.customer_id
ORDER BY c.name;
```

Recent Credit Additions

```sql
SELECT
    ct.created_at,
    c.name as customer,
    ct.amount,
    ct.description
FROM credit_transactions ct
JOIN customers c ON ct.customer_id = c.id
WHERE ct.created_at > NOW() - INTERVAL '7 days'
AND ct.payment_id LIKE 'MANUAL_ADMIN_%'
ORDER BY ct.created_at DESC;
```

🆘 Troubleshooting

"Failed to fetch customers" Error

- **Cause**: Database connection issue or permissions
- **Solution**: Check your database connection and ensure the migration ran successfully

"Forbidden - Admin access required"

- **Cause**: You're not logged in as a super admin
- **Solution**: Log in with a super admin account

Credits not showing in UI

- **Cause**: Migration not run or view not updated
- **Solution**: Run the migration script (`node scripts/run-migration-013.js`)

"Maximum 10,000 credits exceeded"

- **Cause**: Trying to add more than 10,000 credits at once
- **Solution**: Add credits in multiple transactions or modify the limit in the API

📝 Files Modified/Created

Modified Files

1. `/src/app/lib/db/schema.sql` - Updated customer_stats view
2. `/src/app/admin/components/CustomerManagement.tsx` - Added credit UI

New Files

1. `/database/migrations/013_add_credits_to_customer_stats.sql` - Migration file
2. `/database/migrations/APPLY_MIGRATION_013.md` - Migration docs
3. `/scripts/run-migration-013.js` - Migration runner
4. `/ADMIN_CREDIT_MANAGEMENT_GUIDE.md` - This guide

Existing Files (Already Working!)

1. `/src/app/api/admin/credits/add/route.ts` - API endpoint (no changes needed)
2. `/src/app/lib/credits.ts` - Credit management functions (no changes needed)

🎯 Summary

You can now:
- ✅ View credit balances for all customers in the admin dashboard
- ✅ Add credits to any customer account with a simple UI
- ✅ Track all credit additions with full audit trail
- ✅ Use the API programmatically if needed
- ✅ Monitor credit usage and balances

All you need to do:
1. Run the database migration (one-time setup)
2. Log in to the admin dashboard
3. Navigate to Customers
4. Click "💳 Add Credits" on any customer
5. Fill in the amount and description
6. Done!

Perfect for demo accounts, promotional credits, compensations, and testing!

Questions or Issues?
- Check the audit logs in the database
- Review the API endpoint documentation in the code

- Verify your admin permissions
- Ensure the migration ran successfully