Telegram Notifications System - Setup Guide

Overview

This guide explains how to set up and use the Telegram notification system for DeFiDash Tracker, which allows users to receive real-time notifications about Shot Callers posts, whale transactions, Block Wars updates, and market alerts.

Features

🔔 Notification Types

1. **Whale Alerts** - Large cryptocurrency transactions
2. **Shot Callers Posts** - Token calls from selected KOLs
3. **Block Wars Updates** - Game achievements and battles
4. **Market Alerts** - Significant price movements
5. **Daily Summary** - End-of-day market reports

📱 User Experience

- **Back Button**: Automatically shown on settings pages for easy navigation
- **Notification Preferences**: Users can customize which alerts they receive
- **Shot Caller Selection**: Users can follow specific KOLs for notifications
- **Real-time Updates**: Instant delivery of alerts via Telegram

Architecture

Components

1. Telegram Mini App Pages

- **Main Page** (`/telegram-mini/page.tsx`)
- Market data, whale tracking, trending tokens
- Navigation button to notification settings
- Back button hidden on main page

- **Notifications Page** (`/telegram-mini/notifications/page.tsx`)

- Notification preferences UI
- Shot Caller selection interface
- Back button shown for navigation
- Save settings functionality

2. API Endpoints

`/api/telegram/notifications` **(GET/POST)**

Purpose: Manage user notification preferences

GET Request:

```
curl "https://defidashtracker.com/api/telegram/notifications?userId=123456789"
```

Response:

```
{
  "success": true,
  "settings": {
    "whaleAlerts": true,
    "shotCallersAlerts": true,
    "blockWarsAlerts": true,
    "marketAlerts": false,
    "dailySummary": true,
    "selectedShotCallers": ["0xsweep", "100xdarren"]
  }
}
```

POST Request:

```
curl -X POST https://defidashtracker.com/api/telegram/notifications \
  -H "Content-Type: application/json" \
  -d '{
    "userId": 123456789,
    "settings": {
      "whaleAlerts": true,
      "shotCallersAlerts": true,
      "blockWarsAlerts": true,
      "marketAlerts": false,
      "dailySummary": true,
      "selectedShotCallers": ["0xsweep", "100xdarren"]
    }
  }'
```

`/api/telegram/webhook` (POST)

Purpose: Handle incoming messages and commands from Telegram

Supported Commands:
- `/start` - Welcome message
- `/help` - Show available commands
- `/app` or `/miniapp` - Launch Mini App
- `/settings` - Open notification settings
- `/whale` - Get latest whale transaction
- `/alpha` - View alpha feeds
- `/market` - Get market overview

`/api/telegram/notify-shot-callers` (POST)

Purpose: Send notifications when a Shot Caller makes a new post

Usage:

```
curl -X POST https://defidashtracker.com/api/telegram/notify-shot-callers \
  -H "Content-Type: application/json" \
  -d '{
    "username": "0xsweep",
    "displayName": "0xSweep",
    "tokenSymbol": "PEPE",
    "tokenName": "Pepe",
    "sentiment": "bullish",
    "content": "PEPE looking strong! Breaking resistance.",
    "tweetUrl": "https://twitter.com/0xsweep/status/123456789"
  }'
```

Response:

```
{
  "success": true,
  "notificationsSent": 15,
  "message": "Sent notifications to 15 users"
}
```

`/api/shot-callers/list` (GET)

Purpose: Get list of available Shot Callers

Response:

```
{
  "success": true,
  "shotCallers": [
    {
      "username": "0xsweep",
      "displayName": "0xSweep",
      "isVerified": true,
      "profileImage": "/images/0xsweep.jpg"
    }
  ]
}
```

3. Database Schema

The notification settings are stored in the `users` table:

```
model User {
  id                              String    @id @default(cuid())
  email                           String?   @unique
  name                            String?
  telegramChatId                  String?   @unique
  telegramUsername                String?
  telegramNotificationSettings    Json?
  // ... other fields
}
```

Notification Settings JSON Structure:

```
{
  "whaleAlerts": true,
  "shotCallersAlerts": true,
  "blockWarsAlerts": true,
  "marketAlerts": false,
  "dailySummary": true,
  "selectedShotCallers": ["0xsweep", "100xdarren", "cryptowendyo"]
}
```

Setup Instructions

Step 1: Configure Telegram Bot Webhook

Set up the webhook to receive updates from Telegram:

```
curl -X POST "https://api.telegram.org/bot<YOUR_BOT_TOKEN>/setWebhook" \
  -H "Content-Type: application/json" \
  -d '{
    "url": "https://defidashtracker.com/api/telegram/webhook",
    "allowed_updates": ["message", "callback_query"]
  }'
```

Step 2: Set Bot Commands

Configure the bot menu commands:

```
curl -X POST "https://api.telegram.org/bot<YOUR_BOT_TOKEN>/setMyCommands" \
  -H "Content-Type: application/json" \
  -d '{
    "commands": [
      {"command": "start", "description": "Start the bot"},
      {"command": "help", "description": "Show help message"},
      {"command": "app", "description": "Launch Mini App"},
      {"command": "settings", "description": "Notification settings"},
      {"command": "whale", "description": "Latest whale alerts"},
      {"command": "alpha", "description": "Latest alpha feeds"},
      {"command": "market", "description": "Market overview"}
    ]
  }'
```

Step 3: Configure Menu Button

Set the menu button to launch the Mini App:

```
curl -X POST "https://api.telegram.org/bot<YOUR_BOT_TOKEN>/setChatMenuButton" \
  -H "Content-Type: application/json" \
  -d '{
    "menu_button": {
      "type": "web_app",
      "text": "📱 Open App",
      "web_app": {
        "url": "https://defidashtracker.com/telegram-mini"
      }
    }
  }'
```

Step 4: Test the Bot

1. Open Telegram and search for your bot
2. Send `/start` to initialize
3. Send `/app` to launch the Mini App
4. Click the bell icon or use `/settings` to configure notifications

Integration with Shot Callers

Automatic Notifications

When a Shot Caller makes a new post (tweet), call the notification API:

```javascript
// Example: After fetching new tweets from X API
const newTweet = {
  username: "0xsweep",
  displayName: "0xSweep",
  tokenSymbol: "PEPE",
  tokenName: "Pepe",
  sentiment: "bullish",
  content: tweet.text,
  tweetUrl: `https://twitter.com/${tweet.username}/status/${tweet.id}`
};

// Send notification to subscribed users
await fetch('https://defidashtracker.com/api/telegram/notify-shot-callers', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify(newTweet)
});
```

Manual Testing

Test individual notifications:

```bash
# Test whale alert
curl -X POST https://defidashtracker.com/api/telegram/webhook \
  -H "Content-Type: application/json" \
  -d '{
    "update_id": 123456789,
    "message": {
      "message_id": 1,
      "from": {"id": 123456789, "is_bot": false, "first_name": "Test"},
      "chat": {"id": 123456789, "type": "private"},
      "date": 1234567890,
      "text": "/whale"
    }
  }'
```

User Flow

First Time Setup

1. User starts bot with `/start`
2. Bot sends welcome message with Mini App button
3. User launches Mini App

4. User clicks bell icon or Settings button
5. User configures notification preferences
6. User selects Shot Callers to follow
7. User saves settings

Receiving Notifications

1. Shot Caller makes a post on X (Twitter)
2. Your system detects the new post
3. System calls `/api/telegram/notify-shot-callers`
4. API checks which users follow this Shot Caller
5. Notifications sent to subscribed users via Telegram
6. User receives instant notification with post details

Back Button Implementation

The back button is implemented using Telegram WebApp API:

```
useEffect(() => {
  if (window.Telegram?.WebApp) {
    const tg = window.Telegram.WebApp;

    // Show back button on settings page
    tg.BackButton.show();
    tg.BackButton.onClick(() => {
      window.history.back();
    });
  }

  return () => {
    // Hide when leaving page
    if (window.Telegram?.WebApp) {
      window.Telegram.WebApp.BackButton.hide();
    }
  };
}, []);
```

Notification Message Templates

Whale Alert

```
🐋 *Whale Alert*

💰 *Value:* 1,000 ETH (≈ $2,500,000)
⛓ *Blockchain:* ETHEREUM

📤 *From:* `0x123456...abc789`
📥 *To:* `0xabcdef...123456`

🔗 [View Transaction](https://etherscan.io/tx/0x...)
```

Shot Caller Alert

```
🚀 *Alpha Feed - 0xSweep*

💎 *Token:* PEPE (Pepe)
📊 *Sentiment:* BULLISH

💬 "PEPE looking strong! Breaking resistance."

🔗 [View Tweet](https://twitter.com/0xsweep/status/...)
```

Block Wars Update

```
🏆 *Victory!*

You won the battle against CryptoKing!

💰 *Reward:* 1,000 coins
```

Market Alert

```
📈 *Market Alert*

💰 *BTC* (Bitcoin)
💵 *Price:* $42,500
📈 *Change:* +5.23%
📊 *24h Volume:* $25.5M
```

Daily Summary

```
📊 *Daily Market Summary - Oct 24, 2025*

🐋 *Whale Activity*
• Transactions: 45
• Volume: $125.5M

📈 *Top Movers*
• BTC: +5.23%
• ETH: +3.45%
• SOL: +8.91%

💭 *Market Sentiment:* BULLISH

📱 Visit DeFiDash for more insights!
```

Troubleshooting

Notifications Not Received

1. Check user has correct `telegramChatId` in database
2. Verify notification settings are enabled
3. Check Shot Caller username matches subscription
4. Review bot token and webhook configuration

Back Button Not Showing

1. Ensure Telegram WebApp script is loaded
2. Check `tg.BackButton.show()` is called
3. Verify cleanup function hides button properly

Webhook Issues

1. Verify webhook URL is accessible (HTTPS required)
2. Check bot token is correct
3. Review webhook logs for errors
4. Test with `/getWebhookInfo` command

Database Connection

1. Ensure Prisma client is generated
2. Check DATABASE_URL environment variable
3. Verify user permissions for database

Security Considerations

1. **Webhook Validation**: Validate incoming webhook requests from Telegram
2. **Rate Limiting**: Implement rate limits for notification APIs
3. **User Privacy**: Store only necessary user data (chat ID, username)
4. **Token Security**: Keep bot token secure in environment variables
5. **HTTPS**: Always use HTTPS for webhook and Mini App URLs

Future Enhancements

- [ ] Add notification scheduling (quiet hours)
- [ ] Implement notification history
- [ ] Add sound/vibration preferences
- [ ] Group chat support
- [ ] Rich media notifications (images, charts)
- [ ] Notification templates customization
- [ ] Analytics dashboard for notifications
- [ ] A/B testing for notification formats

Support

For issues or questions:
- Check the Telegram Bot API Documentation (https://core.telegram.org/bots/api)
- Review the Mini Apps Documentation (https://core.telegram.org/bots/webapps)
- Contact support at support@defidashtracker.com

Last Updated: October 24, 2025
Version: 1.0.0