

Telegram Integration Guide

Overview

The DeFiDash Smart Money Tracker now has full Telegram bot integration! Users can receive real-time notifications for whale alerts, Block Wars updates, alpha feeds from KOLs, market movements, and daily summaries.

Bot Information

- **Bot Token:** Configured and ready
- **Bot Commands:**
 - `/start` - Initialize the bot and get welcome message
 - `/help` - Show all available commands
 - `/connect` - Get instructions to link your account
 - `/settings` - View your notification preferences
 - `/whale` - Get latest whale transactions
 - `/alpha` - Get latest alpha feeds from Shot Callers
 - `/market` - Get current market overview
 - `/blockwars` - Get your Block Wars stats






Setup for Users

1. Connect Telegram Account

1. Open Telegram and search for your bot username (e.g., `@defidash_bot`)
2. Send `/start` to the bot
3. Send `/connect` to get your Chat ID
4. Go to <https://defidashtracker.com/settings>
5. Navigate to "Notifications" tab
6. Enter your Chat ID and click "Connect Telegram"

2. Configure Notification Preferences

In the Settings page, users can toggle:

-  **Whale Alerts** - Large crypto transfers
-  **Block Wars** - Game updates and achievements
-  **Alpha Feeds** - Token calls from top KOLs
-  **Market Alerts** - Significant price movements
-  **Daily Summary** - End-of-day reports

Developer Integration

Database Schema

The `User` model has been extended with Telegram fields:

```
model User {
  // ... existing fields
  telegramChatId          String? @unique
  telegramUsername        String?
  telegramNotificationSettings Json?
}
```

API Routes

1. Webhook Route (/api/telegram/webhook)

Handles incoming messages from Telegram bot

- Processes commands
- Links user accounts
- Provides interactive responses

2. Send Route (/api/telegram/send)

Sends notifications to users

- POST request with notification type and data
- Filters by user preferences
- Supports batch sending

3. Setup Route (/api/telegram/setup)

Bot configuration and status

- Set webhook URL
- Get bot information
- Verify bot connection

4. User Settings Route (/api/user/telegram-settings)

Manage user Telegram settings

- GET: Fetch user's chat ID and preferences
- POST: Update chat ID and notification settings

Sending Notifications

Import the helper functions from `/lib/telegram-integration-examples.ts` :

```

import {
  sendWhaleAlertNotification,
  sendBlockWarsNotification,
  sendAlphaFeedNotification,
  sendMarketAlertNotification,
  sendDailySummaryNotification
} from '@lib/telegram-integration-examples';

// Example: Send whale alert
await sendWhaleAlertNotification({
  blockchain: 'ethereum',
  value: '100 ETH',
  valueUsd: 350000,
  fromAddress: '0x123...',
  toAddress: '0xabc...',
  txHash: '0xdef...',
});

// Example: Send Block Wars notification
await sendBlockWarsNotification(userId, {
  type: 'win',
  title: 'Battle Victory!',
  description: 'You won the battle and earned 500 coins!',
  reward: 500,
});

```

Integration Points

Whale Tracker

When a large transaction is detected:

```

if (transaction.valueUsd >= WHALE_THRESHOLD) {
  await sendWhaleAlertNotification({
    blockchain: transaction.blockchain,
    value: transaction.value,
    valueUsd: transaction.valueUsd,
    fromAddress: transaction.from,
    toAddress: transaction.to,
    txHash: transaction.hash,
  });
}

```

Block Wars Game

On battle completion or level up:

```
// Battle win
await sendBlockWarsNotification(userId, {
  type: 'win',
  title: 'Victory!',
  description: 'You defeated your opponent!',
  reward: 500,
});

// Level up
await sendBlockWarsNotification(userId, {
  type: 'level_up',
  title: 'Level Up!',
  description: `You reached level ${newLevel}!`,
  reward: 1000,
});

// Achievement
await sendBlockWarsNotification(userId, {
  type: 'achievement',
  title: 'Achievement Unlocked!',
  description: 'Master Trader - Complete 100 battles',
});
```

Shot Callers / KOL Tracking

When a new token call is detected:

```
await sendAlphaFeedNotification({
  kolUsername: kol.username,
  kolDisplayName: kol.displayName,
  tokenSymbol: tokenCall.symbol,
  tokenName: tokenCall.name,
  sentiment: tokenCall.sentiment,
  content: tweet.content,
  tweetUrl: `https://twitter.com/${kol.username}/status/${tweet.id}`,
});
```

Market Monitoring

For significant price changes:

```
if (Math.abs(priceChange) >= 10) {
  await sendMarketAlertNotification({
    symbol: crypto.symbol,
    name: crypto.name,
    priceChange: priceChange,
    currentPrice: crypto.price,
    volume24h: crypto.volume24h,
  });
}
```

Daily Reports

Scheduled at end of day:

```
await sendDailySummaryNotification({
  date: new Date().toLocaleDateString(),
  totalWhaleTransactions: report.totalTransactions,
  totalWhaleVolumeUsd: report.totalVolume,
  topMovers: report.topMovers,
  sentiment: report.sentiment,
});
```

UI Components

TelegramSettings Component

Located at: `/components/telegram/telegram-settings.tsx`

Features:

- Connection status indicator
- Step-by-step setup instructions
- Chat ID input with copy button
- Notification preference toggles
- Test notification button
- Save settings functionality

Settings Page

Located at: `/app/settings/page.tsx`

Integrated tabs:

- Notifications (Telegram settings)
- Profile (placeholder)
- Privacy (placeholder)

Testing

1. Test Bot Connection

```
// GET /api/telegram/setup
// Returns bot information and status
```

2. Set Webhook

```
// POST /api/telegram/setup
// Body: { action: 'set_webhook' }
```

3. Send Test Notification

Use the “Test Notification” button in Settings page or:

```
await fetch('/api/telegram/send', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    type: 'test',
    data: { message: 'Test notification!' },
    userId: 'user-id-here',
  }),
});
```

Security Notes

1. **Bot Token:** Stored securely in `/home/ubuntu/.config/abacusai_auth_secrets.json`
2. **Chat IDs:** Unique per user, stored in database
3. **Webhook:** Only accepts POST requests from Telegram servers
4. **User Verification:** All notifications check user preferences before sending

Monitoring & Logs

Check logs for:

- Webhook processing: `console.log` in webhook route
- Send failures: `console.error` in send route
- Bot errors: Check Telegram bot API responses

Future Enhancements

Potential additions:

- Inline buttons for quick actions
- Interactive polls for market sentiment
- Custom notification thresholds per user
- Group chat support for communities
- Scheduled notifications management
- Multi-language support

Support

For issues or questions:

1. Check bot status: `/api/telegram/setup`
2. Verify webhook: Telegram Bot API dashboard
3. Check user settings: `/api/user/telegram-settings`
4. Test notifications: Settings page test button

Last Updated: October 24, 2025

Version: 1.0

Status:  Production Ready