

Telegram Bot Integration - Complete Guide

Bot Information

- **Bot Name:** Trackerbot
- **Bot Username:** @Tracker103_bot
- **Bot Token:** 8514395374:AAEIhcjlTNAYIbhATI91S0tB1RYrXVVfMzU
- **Status:**  Active and Configured

Integration Overview

The DeFiDash Tracker Bot is fully integrated with your Smart Money Tracker application. All alerts from the system will be sent through this bot.

Integrated Alert Systems

1. Wallet Tracker Alerts

- Monitors watched wallets for new transactions
- Sends real-time notifications for:
 - Sent transactions
 - Received transactions
 - Token swaps
 - Contract interactions
 - **Location:** /app/api/watchlist/check/route.ts
 - **Function:** notifyWalletTransaction()

2. Whale Alerts

- Large crypto transfers
- High-value transactions
- Whale wallet movements
- **API Endpoint:** /api/telegram/whale-feed

3. Shot Callers (Alpha Feeds)

- Token calls from KOLs
- Sentiment analysis
- Tweet notifications
- **API Endpoint:** /api/telegram/notify-shot-callers

4. Market Alerts

- Significant price movements
- Volume spikes
- Market trend changes
- **API Endpoint:** /api/telegram/market-feed

5. Block Wars Notifications

- Battle updates
- Achievement unlocks

- Level-ups
- Rewards

6. Daily Market Summary

- End-of-day reports
- Top movers
- Whale activity summary
- Market sentiment

Available Commands

Users can interact with the bot using these commands:

Command	Description
/start	Start the bot and get welcome message
/help	Show all available commands
/connect	Get your Telegram Chat ID for linking
/app	Launch the Telegram Mini App
/market	Get current market overview
/whale	View latest whale transactions
/alpha	View alpha feeds from KOLs
/settings	Manage notification preferences

Setup Instructions

For Users

1. Find the Bot

- Open Telegram
- Search for `@Tracker103_bot`
- Click "Start" or send `/start`

2. Get Your Chat ID

- Send `/connect` to the bot
- Copy the Chat ID provided
- Go to <https://defidashtracker.com/settings>
- Navigate to "Telegram Notifications"
- Paste your Chat ID and click "Connect"

3. Configure Notifications

- Use `/settings` command in bot

- Or go to app settings page
- Toggle notification preferences:
 -  Whale Alerts
 -  Block Wars
 -  Alpha Feeds
 -  Market Alerts
 -  Daily Summary

For Developers

1. Webhook Setup

```
bash
# Set webhook URL
curl -X POST https://defidashtracker.com/api/telegram/bot-setup
```

2. Test the Bot

```
bash
cd /home/ubuntu/smart_money_tracker/app
npx tsx test-telegram-bot.ts
```

3. Send Test Notification

```
bash
curl -X POST https://defidashtracker.com/api/telegram/send \
-H "Content-Type: application/json" \
-d '{
  "type": "test",
  "data": {
    "message": "Test notification from DeFiDash!"
  }
}'
```



API Endpoints

1. Webhook Handler

Endpoint: POST /api/telegram/webhook

- Receives updates from Telegram
- Processes user commands
- Auto-registers users

2. Bot Setup

Endpoint: POST /api/telegram/bot-setup

- Configures bot commands
- Sets menu button
- Returns bot info

Endpoint: GET /api/telegram/bot-setup

- Get current bot information
- Check bot status

3. Send Notifications

Endpoint: POST /api/telegram/send

```
{
  "type": "whale_alert|block_wars|alpha_feed|market_alert|daily_summary",
  "data": {
    // Type-specific data
  },
  "userId": "optional-user-id"
}
```

4. Notification Feeds

- /api/telegram/whale-feed - Latest whale transactions
- /api/telegram/market-feed - Market data
- /api/telegram/trending-feed - Trending tokens
- /api/telegram/notify-shot-callers - KOL alerts

Security

Bot token is stored securely in:

- **Location:** /home/ubuntu/.config/abacusai_auth_secrets.json
- **Key Path:** telegram.secrets.bot_token.value
- **Access:** Server-side only, never exposed to client

User Settings Storage

Telegram settings are stored in the PostgreSQL database:

```
// User model fields
{
  telegramChatId: String
  telegramUsername: String
  telegramNotificationSettings: {
    whaleAlerts: Boolean
    blockWars: Boolean
    alphaFeeds: Boolean
    marketAlerts: Boolean
    dailySummary: Boolean
    selectedShotCallers: String[]
  }
}
```

Notification Flow

Wallet Tracker Alert Flow

1. Watchlist checker **runs** (/api/watchlist/check)
↓
2. Detects **new** transaction
↓
3. Creates TransactionAlert in **database**
↓
4. Calls **notifyWalletTransaction()**
↓
5. Fetches user's telegram settings
↓
6. Sends **formatted** alert **to** Telegram
↓
7. User receives **notification**

Manual Alert Flow

1. Event occurs in app (whale tx, market move, etc.)
↓
2. API call **to** /api/telegram/send
↓
3. Filters users by **notification** preferences
↓
4. Sends **to** each eligible user
↓
5. **Returns** success/failure report

Message Formatting

All messages use Telegram's Markdown formatting:

```
// Example whale alert
 *Whale Alert*

 *Value:* 1,000 ETH ( $5,128,420)
 *Blockchain:* BASE

 *From:* `0x1234...abcd`
 *To:* `0x5678...efgh`

 [View Transaction](https://bscscan.org/tx/0x...)
```

Testing

Test Bot Connection

```
cd /home/ubuntu/smart_money_tracker/app
npx tsx test-telegram-bot.ts
```

Expected output:

```

✓ Bot connected successfully!
Bot Name: Trackerbot
Username: @Tracker103_bot
Bot ID: 8514395374
✓ Commands set successfully!
✓ Menu button set successfully!

```

Test Wallet Alert

```

import { notifyWalletTransaction } from '@/lib/telegram-client';

await notifyWalletTransaction({
  username: 'your_telegram_username',
  walletAddress: '0x1234...',
  chain: 'base',
  transactionHash: '0xabcd...',
  type: 'received',
  tokenSymbol: 'ETH',
  tokenAmount: '1.5'
});

```

Production Deployment

Webhook Configuration

Set the webhook to your production domain:

```

curl -X POST "https://api.telegram.org/
bot8514395374:AAEihcjLTNAYIbhATI91S0tBlRYrXVVfMzU/setWebhook" \
-d "url=https://defidashtracker.com/api/telegram/webhook"

```

Environment Variables

Ensure these are set in production:

```
NEXT_PUBLIC_APP_URL=https://defidashtracker.com
```

Monitoring

Check Bot Status

```
curl https://defidashtracker.com/api/telegram/bot-setup
```

View Recent Alerts

Check the `TransactionAlert` table in PostgreSQL:

```

SELECT * FROM "TransactionAlert"
ORDER BY "createdAt" DESC
LIMIT 10;

```

Troubleshooting

Bot not responding

1. Check bot token in auth secrets
2. Verify webhook is set correctly
3. Check server logs for errors

Notifications not sending

1. Verify user has `telegramChatId` in database
2. Check notification preferences
3. Ensure bot is not blocked by user

Commands not working

1. Run bot setup: `POST /api/telegram/bot-setup`
2. Clear Telegram cache (close and reopen app)
3. Send `/start` to bot again



Key Files

Backend

- `lib/telegram-client.ts` - Telegram API client
- `app/api/telegram/webhook/route.ts` - Webhook handler
- `app/api/telegram/bot-setup/route.ts` - Bot configuration
- `app/api/telegram/send/route.ts` - Send notifications
- `app/api/watchlist/check/route.ts` - Wallet tracker integration

Frontend

- `components/telegram/telegram-settings.tsx` - Settings UI
- `components/wallet-tracker/transaction-alerts.tsx` - Alert display

Configuration

- `/home/ubuntu/.config/abacusai_auth_secrets.json` - Bot token
- `prisma/schema.prisma` - Database schema



Integration Checklist

- [x] Bot token configured
- [x] Bot commands set
- [x] Menu button configured
- [x] Webhook endpoint created
- [x] `/connect` command for Chat ID
- [x] Wallet tracker alerts integrated
- [x] Whale alerts configured
- [x] Market alerts configured
- [x] Shot Callers alerts configured
- [x] Block Wars alerts configured

- [x] Daily summary configured
- [x] User settings UI created
- [x] Database schema updated
- [x] Test script created

All Systems Ready!

Your Telegram bot `@Tracker103_bot` is fully integrated and ready to send all alerts from the Smart Money Tracker application.

Users can now:

1. Search for `@Tracker103_bot` on Telegram
2. Send `/start` to begin
3. Use `/connect` to get their Chat ID
4. Link their account on defidashtracker.com/settings
5. Receive real-time alerts for all tracked activities!