Social Media Sync - Database Data Structure

Overview

This document explains exactly what data gets saved to the MongoDB database when users sync their social media platforms (TikTok, Twitter, Twitch, Instagram) with HyperBuds.

📊 Data Flow Summary

```
User Clicks "Sync" → Frontend Sends Request → Backend Updates Database → Response Returns

↓ ↓ ↓ ↓

Platform Button {followers: 1000} MongoDB Profile Doc Updated Profile
```

What Gets Sent to Backend

When you click "Sync" on any platform, the frontend sends this minimal data:

Request Format

```
Endpoint: POST /api/v1/profiles/social-sync/{platform}
```

Request Body:

Example for TikTok:

```
POST /api/v1/profiles/social-sync/tiktok
{
    "followers": 1000,
    "engagement": 7.5
}
```

Example for Twitter:

```
POST /api/v1/profiles/social-sync/twitter
{
    "followers": 2500,
    "engagement": 5.8
}
```

Example for Twitch:

```
POST /api/v1/profiles/social-sync/twitch
{
    "followers": 500,
    "engagement": 3.2
}
```

H What Gets Saved in MongoDB

Complete Profile Document Structure

The backend updates your **User Profile Document** in the profiles collection:

```
{
 // DOCUMENT IDENTIFIERS
 " id": "68cec58c9b8985cfa07fe824",
                                   // MongoDB ObjectId (Profile ID)
 "userId": "68cec2a59b8985cfa07fe80a",
                                   // Reference to User document
 // BASIC PROFILE INFORMATION
 "displayName": "Yash Gupta",
                                   // User's display name
 "bio": "Content creator focused on tech", // User biography
 "avatar": "https://example.com/avatar.jpg", // Profile picture URL
 "profileUrl": "https://hyperbuds.com/yash", // Profile URL
 "niche": [
                                    // Creator niches/categories
   "music".
   "education",
   "gaming",
   "comedy",
   "fitness"
 1,
 "rizzScore": 79,
                                    // Rizz score (0-100)
 "isPublic": true,
                                    // Profile visibility
 "isActive": true,
                                    // Account active status
 // PLATFORM STATISTICS (THE IMPORTANT PART!)
 "stats": {
   // Individual platform breakdowns
   "platformBreakdown": {
    // TikTok Data
    "tiktok": {
      "followers": 1000, // ✓ SYNCED: TikTok follower count
      "engagement": 7.5 // ✓ SYNCED: TikTok engagement rate (%)
    },
    // Instagram Data
    "instagram": {
      "followers": 15925, // ☑ SYNCED: Instagram follower count
      "engagement": 2.91 // ✓ SYNCED: Instagram engagement rate (%)
    },
    // YouTube Data
    "youtube": {
```

```
"followers": 7100, // ✓ SYNCED: YouTube subscriber count
    "engagement": 2.66
                    // ☑ SYNCED: YouTube engagement rate (%)
  },
  // Twitch Data
  "twitch": {
    "followers": 0, // ✓ SYNCED: Twitch follower count
    "engagement": 0
                    // ☑ SYNCED: Twitch engagement rate (%)
  }
 },
 // Automatically calculated aggregates
 "totalFollowers": 24025, // ✓ AUTO-CALCULATED: Sum of all platforms
 "avgEngagement": 1.39 // ✓ AUTO-CALCULATED: Average engagement
},
// USER PREFERENCES
"preferences": {
 "audienceSize": {
   "min": 0,
  "max": 1000000
 },
 "budget": {
  "min": 0,
  "max": 10000
 "collaborationTypes": [],
 "locations": []
},
// LOCATION INFORMATION
"location": {
 "city": "Delhi",
 "state": "Delhi",
 "country": "India"
},
// TIMESTAMPS
"lastSeen": "2025-09-26T21:02:18.953Z", // Last activity timestamp
"createdAt": "2025-09-20T15:17:32.306Z", // Profile creation date
```

```
// METADATA
"__v": 0
        // MongoDB version key
```



Exact Fields Updated During Sync

When You Sync TikTok:

Database Field Path	What Gets Saved	Example Value	Data Type
stats.platformBreakdown.tiktok.followers	Your TikTok follower count	1000	Number
stats.platformBreakdown.tiktok.engagement	Your TikTok engagement rate	7.5	Number
stats.totalFollowers	Recalculated total followers	24025	Number
stats.avgEngagement	Recalculated average engagement	1.39	Number
updatedAt	Current timestamp	"2025-10-08T19:01:19.222Z"	Date

When You Sync Twitter:

Database Field Path	What Gets Saved	Example Value	Data Type
stats.platformBreakdown.twitter.followers	Your Twitter follower	2500	Number

Database Field Path	What Gets Saved	Example Value	Data Type
	count		
stats.platformBreakdown.twitter.engagement	Your Twitter engagement rate	5.8	Number
stats.totalFollowers	Recalculated total followers	26525	Number
stats.avgEngagement	Recalculated average engagement	4.02	Number
updatedAt	Current timestamp	"2025-10-08T19:05:32.145Z"	Date

When You Sync Twitch:

Database Field Path	What Gets Saved	Example Value	Data Type
stats.platformBreakdown.twitch.followers	Your Twitch follower count	500	Number
stats.platformBreakdown.twitch.engagement	Your Twitch engagement rate	3.2	Number
stats.totalFollowers	Recalculated total followers	27025	Number
stats.avgEngagement	Recalculated average engagement	3.8	Number
updatedAt	Current timestamp	"2025-10-08T19:10:15.782Z"	Date

Step-by-Step Database Update Process

```
// STEP 1: Validate Request
const { platform } = req.params; // e.g., "tiktok"
const { followers, engagement } = req.body;
const userId = req.user.id;
// Validate platform type
const validPlatforms = ['tiktok', 'twitter', 'twitch', 'instagram'];
if (!validPlatforms.includes(platform)) {
 throw new Error('Invalid platform');
}
// Validate data
if (typeof followers !== 'number' || followers < 0) {</pre>
 throw new Error('Invalid followers count');
}
// STEP 2: Update Platform-Specific Data
const profile = await Profile.findOneAndUpdate(
 { userId: userId },
 {
   $set: {
    [`stats.platformBreakdown.${platform}.followers`]: followers,
    [`stats.platformBreakdown.${platform}.engagement`]: engagement | | 0,
    updatedAt: new Date()
   }
 },
 { new: true } // Return updated document
);
// STEP 3: Recalculate Aggregate Statistics
const platformBreakdown = profile.stats.platformBreakdown;
// Calculate total followers across all platforms
const totalFollowers =
 (platformBreakdown.tiktok?.followers | 0) +
```

```
(platformBreakdown.instagram?.followers | 0) +
 (platformBreakdown.youtube?.followers | 0) +
 (platformBreakdown.twitch?.followers | 0);
// Calculate average engagement across all platforms
const engagements = [
 platformBreakdown.tiktok?.engagement | 0,
 platformBreakdown.instagram?.engagement | 0,
 platformBreakdown.youtube?.engagement | 0,
 platformBreakdown.twitch?.engagement | 0
].filter(e \Rightarrow e > 0);
const avgEngagement = engagements.length > 0
 ? engagements.reduce((sum, e) => sum + e, 0) / engagements.length
 : 0;
// STEP 4: Update Aggregates in Database
await Profile.findByIdAndUpdate(profile._id, {
 $set: {
   'stats.totalFollowers': totalFollowers,
   'stats.avgEngagement': avgEngagement
 }
});
// STEP 5: Return Updated Profile
const updatedProfile = await Profile.findById(profile. id);
return {
 success: true,
 message: `Social media ${platform} synced successfully`,
 profile: updatedProfile
};
```

👔 Data Examples

Example 1: Fresh User (No Previous Data)

Before Sync:

```
{
    "stats": {
        "platformBreakdown": {
            "tiktok": { "followers": 0, "engagement": 0 },
            "instagram": { "followers": 0, "engagement": 0 },
            "youtube": { "followers": 0, "engagement": 0 },
            "twitch": { "followers": 0, "engagement": 0 }
        },
        "totalFollowers": 0,
        "avgEngagement": 0
    }
}
```

User Syncs TikTok with 1000 followers, 7.5% engagement:

After Sync:

```
{
   "stats": {
      "platformBreakdown": {
           "tiktok": { "followers": 1000, "engagement": 7.5 },
            "instagram": { "followers": 0, "engagement": 0 },
            "youtube": { "followers": 0, "engagement": 0 },
            "twitch": { "followers": 0, "engagement": 0 }
    },
    "totalFollowers": 1000,
    "avgEngagement": 7.5
}
```

Example 2: Existing User (Multiple Platforms)

Before Sync:

```
{
   "stats": {
      "platformBreakdown": {
        "tiktok": { "followers": 1000, "engagement": 7.5 },
        "instagram": { "followers": 15925, "engagement": 2.91 },
        "youtube": { "followers": 7100, "engagement": 2.66 },
        "twitch": { "followers": 0, "engagement": 0 }
    },
      "totalFollowers": 24025,
      "avgEngagement": 4.36
}
```

User Syncs Twitch with 500 followers, 3.2% engagement:

After Sync:

```
{
    "stats": {
        "platformBreakdown": {
            "tiktok": { "followers": 1000, "engagement": 7.5 },
            "instagram": { "followers": 15925, "engagement": 2.91 },
            "youtube": { "followers": 7100, "engagement": 2.66 },
            "twitch": { "followers": 500, "engagement": 3.2 }
        },
        "totalFollowers": 24525,
        "avgEngagement": 4.07
    }
}
```

Example 3: Re-Syncing Updated Data

Before Sync:

```
{
  "stats": {
    "platformBreakdown": {
        "tiktok": { "followers": 1000, "engagement": 7.5 }
    },
    "totalFollowers": 1000,
    "avgEngagement": 7.5
  }
}
```

User Re-Syncs TikTok with 2000 followers, 8.2% engagement:

After Sync:

```
{
   "stats": {
     "platformBreakdown": {
        "tiktok": { "followers": 2000, "engagement": 8.2 }
     },
     "totalFollowers": 2000,
     "avgEngagement": 8.2
   }
}
```

What IS Saved

Data	Saved?	Location in Database
Follower count	✓ Yes	stats.platformBreakdown.{platform}.followers
Engagement rate	✓ Yes	stats.platformBreakdown.{platform}.engagement
Total followers (calculated)	✓ Yes	stats.totalFollowers
Average engagement (calculated)	✓ Yes	stats.avgEngagement
Last sync timestamp	✓ Yes	updatedAt

X What is NOT Saved

Data	Saved?	Why Not?
Individual posts or content	× No	Not needed for matching
Follower names or details	× No	Privacy concerns
Post history	× No	Not needed for matching
Platform credentials/passwords	× No	Security risk
Personal DMs or messages	× No	Privacy concerns

Data	Saved?	Why Not?
Video/image content	× No	Storage constraints
Comments or interactions	× No	Not needed for matching
Platform analytics details	× No	Only summary stats needed

-

Database Information

Collection Details

Property	Value
Database Name	hyperbuds
Collection Name	profiles
Database Type	MongoDB (NoSQL)
Hosting	MongoDB Atlas (Production)
Schema Type	Flexible (Mongoose Schema)

Indexes

```
// Performance indexes
profiles.createIndex({ userId: 1 }, { unique: true });
profiles.createIndex({ 'stats.totalFollowers': -1 });
profiles.createIndex({ 'stats.avgEngagement': -1 });
profiles.createIndex({ niche: 1 });
profiles.createIndex({ isActive: 1, isPublic: 1 });
```

Document Size

Metric	Typical Value
Average document size	~2-3 KB
Maximum document size	16 MB (MongoDB limit)
Typical number of fields	15-20

o Why This Data Matters

Use Cases

1. Matching Algorithm

- Find compatible creators based on follower counts
- Match by engagement rates
- · Filter by total audience size

2. Profile Display

- Show verified follower counts to potential collaborators
- Display engagement statistics
- · Build credibility with real data

3. Analytics & Insights

- · Track follower growth over time
- Monitor engagement trends
- · Compare platform performance

4. Search & Filtering

- Other users can filter creators by audience size
- Search by platform presence
- Find creators in specific follower ranges

5. Credibility & Trust

- · Verified data builds trust
- · Real numbers prevent fake profiles
- Transparent statistics



Data Source

The follower and engagement data comes from:

Production (Real Data)

- RapidAPI Integration Live data from social platforms
- Platforms supported: TikTok, Twitter, Twitch, Instagram
- · Real-time fetching when user clicks sync

Development (Mock Data)

Mock Data Generator - Fake data for testing

- Predictable test values
- · No external API calls required

Security & Privacy

Data Protection

1. Authentication Required

- All sync endpoints require JWT token
- User can only sync their own profile
- No access to other users' sync endpoints

2. Data Validation

- Followers must be positive numbers
- Engagement must be 0-100%
- · Input sanitization on backend

3. Privacy Controls

- Users control profile visibility (isPublic)
- Can choose which platforms to sync
- Can re-sync to update or remove data

Data Retention

- Forever Data persists until user deletes account
- Updates Can re-sync anytime to update values
- **Deletion** Removed when account is deleted

Total Followers Calculation

```
totalFollowers =
  tiktok.followers +
  instagram.followers +
  youtube.followers;

// Example:
// TikTok: 1000
// Instagram: 15925
// YouTube: 7100
// Twitch: 500
// Total: 24525
```

Average Engagement Calculation

Testing Data

Test Scenarios

```
// Scenario 1: New user, first sync
{
    "before": { totalFollowers: 0, avgEngagement: 0 },
    "sync": { platform: "tiktok", followers: 1000, engagement: 7.5 },
    "after": { totalFollowers: 1000, avgEngagement: 7.5 }
}

// Scenario 2: Existing user, additional platform
{
    "before": { totalFollowers: 1000, avgEngagement: 7.5 },
    "sync": { platform: "instagram", followers: 5000, engagement: 3.2 },
    "after": { totalFollowers: 6000, avgEngagement: 5.35 }
}

// Scenario 3: Re-sync existing platform
{
    "before": { "tiktok.followers": 1000 },
    "sync": { platform: "tiktok", followers: 2000, engagement: 8.0 },
    "after": { "tiktok.followers": 2000 }
}
```

Related Documentation

- Backend Integration Guide
- API Specification
- Frontend Implementation
- Testing Guide
- Architecture Overview

Last Updated: October 10, 2025

Document Version: 1.0

Maintained By: HyperBuds Development Team