

APBeeper Bot Setup Complete!

Your multi-server APB Discord bot has been successfully created and is ready for deployment.

What's Been Created

Core Architecture

- **Multi-server support** with SQLite database (easily upgradeable to PostgreSQL/MySQL)
- **Per-server settings** storage for game names, channels, clan roles
- **OAuth2 invite system** for easy server addition
- **Clean, modular code structure** with proper error handling

Features Implemented

Discord Status Tracking

- `/setgame <game_name>` - Set the game to track
- `/players` - Show current players online
- `/setclangroup <role>` - Set clan role for member separation
- Automatically separates clan members from community members
- Displays clan members at the top

APB Population Display

- `/apbpop [region]` - Show current APB population (NA/EU/Both)
- `/setchannel <channel> [region]` - Set up auto-updating panels
- Clean visual display with Crim/Enforcer breakdown
- Auto-updates every 5 minutes
- Hides empty districts, shows only populated ones

Twitch Integration

- `/twitch add <url>` - Users can add their streams
- `/twitch remove` - Remove your stream
- `/twitch list` - List all streamers
- **Admin controls:**
 - `/twitch admin enable/disable` - Toggle features
 - `/twitch admin add <user> <url>` - Add stream for user
 - `/twitch admin remove <user>` - Remove user's stream
 - `/twitch admin setchannel <channel>` - Set notification channel
- Stream alerts with preview (no pings)
- Automatic live status tracking

Technical Features

- **Database schema** with proper indexing
- **Slash commands** with permission controls
- **Auto-updating message system** with error recovery
- **Comprehensive logging** with file output
- **Clean embed styling** with consistent branding

- **Scheduled tasks** for population updates and stream checking

Quick Start

1. Configure Environment

```
bash
# Edit .env file with your credentials
nano .env
```

Add your Discord bot token and client ID.

2. Start the Bot

```
bash
npm start
```

3. Invite to Server

Use this URL template (replace CLIENT_ID):

```
https://discord.com/api/oauth2/authorize?cli-
ent_id=CLIENT_ID&permissions=274877975552&scope=bot%20applications.commands
```

Project Structure

```
apbeeper_bot/
├── src/
│   ├── index.js           # Main bot entry point
│   ├── database.js        # Database connection and queries
│   ├── commands/         # Slash command handlers
│   │   ├── apbpop.js      # APB population display
│   │   ├── players.js     # Discord status tracking
│   │   ├── setgame.js     # Set tracked game
│   │   ├── setclangroup.js # Set clan role
│   │   ├── setchannel.js  # Set APB update channel
│   │   └── twitch.js      # Twitch integration
│   ├── services/         # Core services
│   │   ├── discordStatus.js # Discord status tracking
│   │   ├── apbPopulation.js # APB population fetching
│   │   └── twitchNotify.js  # Twitch integration
│   └── utils/            # Utilities
│       ├── embedStyles.js # Embed styling
│       └── logger.js      # Logging utility
├── data/                # Database storage
├── logs/                # Log files
├── package.json          # Dependencies and scripts
├── .env.example          # Environment template
├── .gitignore            # Git ignore rules
└── README.md             # Detailed documentation
```

Configuration Notes

Required Permissions

The bot needs these Discord permissions:

- Send Messages
- Use Slash Commands

- Embed Links
- Read Message History
- Manage Messages (for auto-updating panels)
- View Channels

Environment Variables

- `DISCORD_TOKEN` - Your Discord bot token (required)
- `CLIENT_ID` - Your Discord application client ID (required)
- `TWITCH_CLIENT_ID` - Twitch API client ID (optional)
- `TWITCH_CLIENT_SECRET` - Twitch API client secret (optional)

Database Upgrade Path

To upgrade from SQLite to PostgreSQL/MySQL:

1. Install appropriate database driver
2. Update connection in `src/database.js`
3. Update connection string in environment variables

Key Features

Multi-Server Ready

- No manual configuration per server
- Automatic default settings creation
- Per-server customization through commands

Production Ready

- Comprehensive error handling
- Automatic recovery from message deletions
- Rate limiting protection
- Structured logging

Extensible Design

- Modular command structure
- Service-based architecture
- Easy to add new features
- Clean separation of concerns

Support

For issues or questions:

1. Check the logs in `logs/` directory
2. Review the README.md for detailed documentation
3. Ensure all environment variables are properly set
4. Verify bot permissions in Discord servers

Your APBeeper bot is now ready to serve multiple APB gaming communities!