

Complete Discord Server Setup Guide

Overview

This expanded Discord MCP server can now set up entire Discord servers from natural language descriptions. It includes 50+ tools covering every aspect of Discord server management.

Key Features Added



AI-Driven Server Setup

- **Natural Language Processing:** Describe your server in plain English
- **Template-Based Setup:** Pre-built templates for different server types
- **Intelligent Channel/Role Creation:** Automatically creates appropriate structure
- **Content Generation:** Auto-generates rules, welcome messages, and descriptions



Comprehensive Server Management

Server Settings

- `edit_server_settings` - Complete server configuration
- `create_server_template` - Generate reusable templates
- `setup_complete_server` - AI-driven full setup

Advanced Channel Management

- `create_channel_category` - Organize channels
- `create_voice_channel` - Voice chat rooms
- `create_stage_channel` - Presentation/lecture channels
- `create_forum_channel` - Discussion forums
- `create_announcement_channel` - News/announcements
- `edit_channel` - Modify existing channels
- `set_channel_permissions` - Granular permission control

Role & Permission System

- `create_role` - Detailed role creation
- `edit_role` - Modify existing roles
- `delete_role` - Remove roles

- `create_role_hierarchy` - Multi-role setup
- Advanced permission management

Content & Customization

- `create_emoji` - Custom server emojis
- `create_sticker` - Custom stickers
- `create_webhook` - External integrations
- `send_webhook_message` - Automated posting

Moderation & Safety

- `create_automod_rule` - Automated content filtering
- `ban_member` - Ban users with options
- `kick_member` - Remove users
- `timeout_member` - Temporary restrictions
- `bulk_delete_messages` - Mass message cleanup

Community Features

- `create_scheduled_event` - Server events
- `create_invite` - Customizable invites
- `create_thread` - Discussion threads

Usage Examples

1. Complete Server Setup from Description

```
python
```

Example AI prompt to Claude:

"""

Set up a gaming server called "Elite Esports Hub" for competitive gaming.

We need:

- Separate sections for different games (Valorant, CS2, League of Legends)
- Voice channels for team practice
- Tournament announcement channels
- Role hierarchy (Owner, Admin, Coach, Player, Spectator)
- Family-friendly environment with strict moderation
- Forum channels for strategy discussions
- Stage channel for coaching sessions

"""

This would call:

```
setup_complete_server(  
    server_id="123456789",  
    server_description="Gaming server for competitive esports...",  
    server_type="gaming"  
)
```

2. Business/Corporate Server Setup

python

Natural language description:

"""

Create a professional business server for our tech startup "InnovateCorp".

We need departments for Engineering, Marketing, Sales, and HR.

Include meeting rooms, project channels, and strict permission controls.

Only employees should see internal channels.

"""

Results in structured setup with:

- # - Department-specific channels
- # - Permission-controlled access
- # - Professional role hierarchy
- # - Meeting voice channels
- # - Project management areas

3. Educational Server Setup

python

Example description:

|||||

Set up an educational server for "Advanced Computer Science Course".
Include lecture halls, study groups, assignment submission areas,
and Q&A forums. Need roles for Professor, TA, and Students.

|||||

Creates:

- Lecture stage channels

- Study voice rooms

- Assignment text channels

- Q&A forum channels

- Academic role structure

Server Type Templates

Gaming Servers

- **Categories:** Info, General Chat, Gaming, Voice, Admin
- **Special Features:** Game-specific channels, LFG areas, stat tracking
- **Roles:** Owner, Admin, Moderator, Gamer, VIP, Member
- **Voice:** Multiple gaming voice channels, stage for announcements

Business Servers

- **Categories:** Company Info, Departments, Meetings, Projects, Admin
- **Special Features:** Department channels, meeting rooms, project tracking
- **Roles:** CEO, Management, Employee, Contractor
- **Professional:** Conference rooms, all-hands stage channel

Educational Servers

- **Categories:** Course Info, Classes, Study Groups, Lectures, Staff
- **Special Features:** Lecture halls, study rooms, assignment areas
- **Roles:** Instructor, Teaching Assistant, Student, Auditor
- **Academic:** Stage for lectures, voice study rooms

Creative Communities

- **Categories:** Showcase, Collaboration, Resources, Voice, Admin

- **Special Features:** Portfolio channels, collaboration spaces
- **Roles:** Artist, Collaborator, Critic, Member
- **Creative:** Showcase galleries, collaboration voice channels

General Communities

- **Categories:** Welcome, General, Topics, Voice, Admin
- **Special Features:** Discussion forums, community events
- **Roles:** Moderator, Active Member, Member, New Member
- **Social:** Community voice channels, event spaces

Advanced Features

AutoModeration Setup

The system automatically creates moderation rules based on your server description:

```
python

# Example automod rules created:
{
  "name": "Anti-Spam Protection",
  "trigger_type": "spam",
  "actions": ["block_message", "timeout"]
}

{
  "name": "Family Friendly Filter",
  "trigger_type": "keyword_preset",
  "keyword_presets": ["profanity", "sexual_content"],
  "actions": ["block_message", "alert_moderators"]
}
```

Role Hierarchy Management

Automatically creates appropriate role structures:

```
python
```

Gaming server roles (in hierarchy order):

- "👑 Server Owner" (Administrator)
- "🛡️ Admin" (Administrator)
- "🔨 Moderator" (Kick, Ban, Manage Messages)
- "🎮 Gamer" (Send Messages, Connect)
- "🌟 VIP" (Send Messages, Connect)
- "👤 Member" (Basic permissions)

📋 Channel Organization

Intelligent channel creation with proper categorization:

python

Example structure:

📋 Information/

- ├── 📖 -rules
- ├── 📢 -announcements
- └── ❓ -support

💬 General Chat/

- ├── 💬 -general
- ├── 🎯 -introductions
- └── 🌍 -off-topic

🎮 Gaming/

- ├── 🎮 -general-gaming
- ├── 🎯 -lfg
- └── 🎮 -game-forum (Forum Channel)

🔊 Voice Channels/

- ├── 🔊 General Voice
- ├── 🎮 Gaming Voice 1
- └── 🎤 Stage Channel (Stage)

Integration Instructions

1. Install Dependencies

bash

```
# Add to requirements.txt
discord.py>=2.4.0
mcp>=0.1.0
aiohttp>=3.8.0
```

2. Update Main Server File

Replace your `server.py` with the expanded version and add the AI setup logic:

```
python

# Add to imports
from server_setup_templates import setup_server_from_description, execute_setup_plan

# In the setup_complete_server tool handler:
elif name == "setup_complete_server":
    server_id = arguments["server_id"]
    description = arguments["server_description"]
    server_type = arguments.get("server_type", "general")

    # Generate setup plan using AI logic
    plan = setup_server_from_description(server_id, description, server_type)

    # Execute the plan
    results = await execute_setup_plan(discord_client, server_id, plan)

    return [TextContent(
        type="text",
        text=f"🤖 AI Server Setup Complete!\n\n" + "\n".join(results)
    )]
```

3. Enable Required Permissions

Update your bot's permissions to include:

- ✓ Manage Server
- ✓ Manage Roles
- ✓ Manage Channels
- ✓ Manage Messages
- ✓ Kick Members
- ✓ Ban Members
- ✓ Create Instant Invite
- ✓ Manage Webhooks
- ✓ Manage Emojis and Stickers
- ✓ Use Slash Commands
- ✓ Manage Events
- ✓ Moderate Members

4. Update Claude Desktop Config

```
json
{
  "mcpServers": {
    "discord": {
      "command": "uv",
      "args": [
        "--directory",
        "/path/to/mcp-discord",
        "run",
        "mcp-discord"
      ],
      "env": {
        "DISCORD_TOKEN": "your_bot_token_here"
      }
    }
  }
}
```

Example Conversations with Claude

Gaming Server Setup

User: "Set up a competitive FPS gaming server with separate areas for Valorant and CS2, voice channels for teams, and a tournament system."

Claude: I'll set up a comprehensive gaming server for you! Let me create:

1. Game-specific categories and channels
2. Team voice channels
3. Tournament announcement areas
4. Competitive role hierarchy
5. Moderation rules for fair play

[Uses setup_complete_server tool with detailed gaming template]

Business Server Setup

User: "Create a professional workspace for our 50-person marketing agency with department channels and meeting rooms."

Claude: I'll create a professional business environment with:

1. Department-specific channels (Creative, Strategy, Accounts, Admin)
2. Client project channels with proper permissions
3. Meeting voice channels and conference rooms
4. Professional role structure
5. Business-appropriate moderation

[Creates structured business server layout]

Educational Server Setup

User: "Build a university course server for my Computer Science class with 200 students."

Claude: I'll set up an academic environment featuring:

1. Lecture hall stage channel for presentations
2. Study group voice rooms
3. Assignment submission channels
4. Q&A forum for student questions
5. Academic role hierarchy (Professor, TAs, Students)
6. Study-focused moderation rules

[Implements educational template with academic features]

Monitoring and Management

Built-in Analytics

- Track setup success rates
- Monitor channel usage

- Role assignment analytics
- Moderation effectiveness

Maintenance Tools

- Bulk permission updates
- Channel reorganization
- Role hierarchy adjustments
- Automated cleanup routines

Best Practices

1. Server Planning

- Clearly describe your community's purpose
- Specify expected member count
- Define moderation requirements
- Plan for growth and scaling

2. Permission Management

- Use principle of least privilege
- Regular permission audits
- Role hierarchy maintenance
- Channel-specific restrictions

3. Community Guidelines

- Clear, enforceable rules
- Consistent moderation
- Regular community feedback
- Adaptation to growth

4. Automation Balance

- Automate routine tasks
- Maintain human oversight
- Regular rule adjustments
- Community input on changes

Troubleshooting

Common Issues

1. **Permission Errors:** Ensure bot has Administrator permissions
2. **Channel Limits:** Discord has limits on channels per server
3. **Role Limits:** Maximum 250 roles per server
4. **Rate Limiting:** Large setups may hit API rate limits

Solutions

- Implement error handling and retry logic
- Batch operations with delays
- Graceful degradation for limits
- Clear error reporting to users

This comprehensive system transforms Discord server setup from a manual, time-consuming process into an intelligent, automated workflow that can create professional-grade communities in minutes rather than hours.