

Runbook: Setting Up NextPlay Financial Discord Bot

Part 1 — Discord Setup

1. Create a Discord Server

1. Open Discord → bottom left sidebar → “+” (**Add a Server**).
 2. Choose **Create My Own**.
 3. Give it a name (e.g., *NextPlay Financial*).
 4. You’ll be the **Owner** automatically.
-

2. Create Bot Application in Developer Portal

1. Go to Discord Developer Portal.
2. Click “**New Application**” → name it *NextPlay Financial Bot*.
3. In left sidebar, go to **Bot** tab.
4. Click “**Add Bot**” → confirm.
5. Under **Token**: click **Reset Token** → copy the value (this is your DISCORD_BOT_TOKEN).

⚠ Security note: Never share this token. Treat it like a password.

3. Configure Bot Permissions

1. Still under **Bot** tab → enable:
 - **Privileged Gateway Intents:**
 - ☒ Message Content Intent
 - ☒ Server Members Intent
 - ☒ Presence Intent (optional but good for monitoring).
2. In sidebar → **OAuth2** → **URL Generator**:
 - Under **Scopes**: select bot and applications.commands.

- Under **Bot Permissions**:
 - ☒ Read Messages/View Channels
 - ☒ Send Messages
 - ☒ Attach Files
 - ☒ Read Message History
 - ☒ Use Slash Commands

3. Copy the generated **OAuth2 URL** → paste in browser → invite bot to your server.

4. Create Dedicated Channels

Inside your server:

- Create a **text channel** called #chat (for /chat).
 - Create a **text channel** called #coach (for /coach).
-

Part 2 — GitHub and Code

1. Push Code to GitHub

1. Initialize a Git repo:
2. `git init`
3. `git remote add origin https://github.com/<your-org>/<repo>.git`
4. `git add .`
5. `git commit -m "Initial commit"`
6. `git push origin main`
7. Repo will contain:
 - `main.py`
 - `services/openai_coach.py`
 - `requirements.txt`

- README.md (with summary info)
 - .env.example
-

2. Example .env

Provide .env file (not committed to repo):

DISCORD_BOT_TOKEN=xxxxxxxxxxxxxxxxxxxxxx

OPENAI_API_KEY=xxxxxxxxxxxxxxxxxxxxxx

NPF_ASSISTANT_ID=xxxxxxxxxxxxxxxxxxxxxx

CHAT_CHANNEL=chat

COACH_CHANNEL=coach

RATE_LIMIT_PER_MINUTE=5

Part 3 — Render Deployment

1. Create Service

1. Go to Render Dashboard.
 2. **New → Web Service → Connect GitHub Repo.**
 - Pick your bot's repo.
 3. Service Type → **Background Worker** (not Web Service).
-

2. Environment

- Runtime: **Python 3.11+**
 - Start Command:
 - python main.py
 - Add all environment variables from .env.
-

3. Reliability

- Render automatically restarts crashed workers.
 - Subscribed to **\$7 Starter** plan = sufficient for 24/7 bot uptime.
-

Part 4 — OpenAI Assistants API Setup

1. Create Assistant

1. Go to [OpenAI Platform](#).
 2. Navigate to **Assistants**.
 3. Create new Assistant → name it **NextPlay Coach**.
 4. Select **Model**: gpt-4o-mini (fast) or gpt-5 (deep Q&A).
 5. Add **Tools**:
 - ☒ File Search
 6. Save.
 7. Copy the **Assistant ID** (starts with asst_...). → Put into .env as NPF_ASSISTANT_ID.
-

2. Assistant Prompt

Set the **system prompt** for your Assistant:

You are NextPlay Coach, a professional financial education-only assistant for Discord.

Core behavior:

- Always answer strictly from the user's uploaded documents using retrieval/file_search.
- For targeted Q&A:
 - Return the EXACT QUOTED sentence(s) from the source before any explanation.
 - Always include filename.
 - Do not summarize unless the user explicitly asks to “summarize”, “overview”, or “tl;dr”.
- If no direct answer exists in the files, reply exactly: “I could not find that in the provided documents.”

Citations:

- After every answer, add a **Citations** section with numbered references [1], [2], etc.
- Each citation must be formatted as: [1] "quoted text..." (Filename.pdf, page X)

Guardrails:

- Never provide allocations, percentages, buy/sell instructions, or product recommendations.
- If asked for those, reply: "I can't recommend specific allocations or trades, but I can explain how investors generally think about diversification."

Tone & format:

- Professional, neutral, concise.
- Use short bullet points or brief paragraphs.
- Always append this disclaimer line:

This information is for educational purposes only and not financial advice. Please consult a licensed financial professional before making any investment decisions.

Part 5 — Local Testing (Optional)

Run locally before deployment:

```
pip install -r requirements.txt
```

```
python main.py
```

- Join your server.
- Test:
 - /chat What is diversification?
 - /coach → upload PDF → ask targeted question.
- /reset clears memory.

Part 6 — Final Verification

1. Deploy to Render.
2. Join server with bot.
3. Upload a known PDF → ask:
4. /coach What is the first example in the document?

→ Should return **exact quote + filename + page number**.

5. Test empty/corrupted PDF.
6. Confirm every reply ends with disclaimer.