



Setting up RoMod for Roblox and Discord

RoMod is a Discord bot that brings moderation and game management tools to your Roblox experience. This tutorial walks you through inviting the bot, connecting it to your Roblox game with an Open Cloud API key, enabling custom and advanced commands, and testing that everything works with a live moderation command.

By the end of this guide, you'll have a fully configured RoMod setup that lets you moderate your Roblox game — including kicking and banning players — directly from Discord.

Prerequisites

Before you begin, make sure you have:

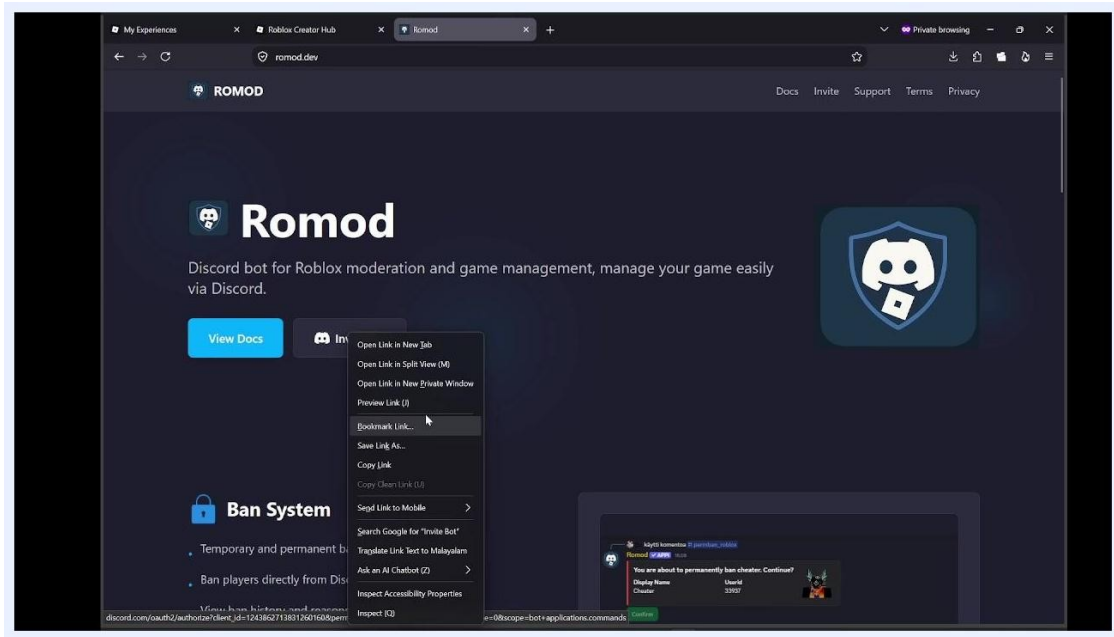
- Administrator permissions in your Discord server.
- Developer permissions in Roblox Studio.
- The following open in your browser:
 -
- Your Discord server
-

Step 1: Invite the RoMod bot to your Discord server

1. Go to [this link](#) in your web browser.
2. Locate the Invite link for the RoMod bot.



3. Right-click the invite link and select Copy Link.



RoMod website with invite link context menu open, showing options like "Open Link in New Tab", "Copy Link", and "Bookmark Link"

4. Paste the copied invite link into your Discord server's appropriate channel.
5. Follow the Discord prompts to add the RoMod bot to your server, granting the necessary permissions as requested by Discord.

Step 2: Begin RoMod setup in Discord

In your Discord server, type the command:

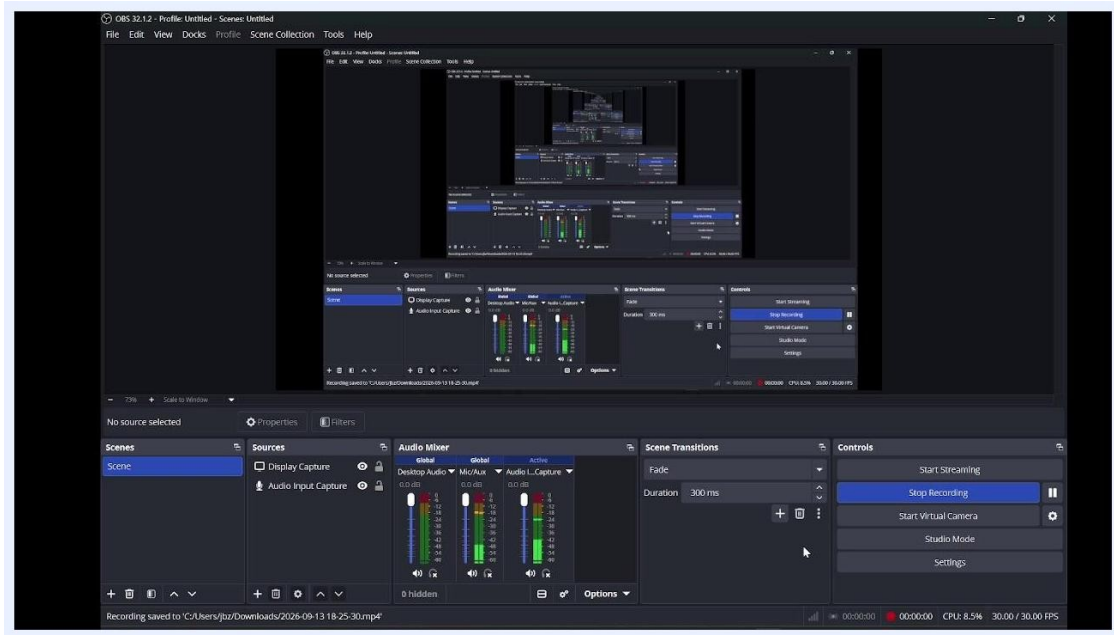
```
/setup
```

The bot will prompt you to obtain an API key from Roblox. You're ready to move over to the Roblox Creator Hub to generate that key.



Step 3: Generate a Roblox Open Cloud API key

6. Go to [Roblox Studio](#) and log in.
7. In the left sidebar, click Creations to view your experiences.

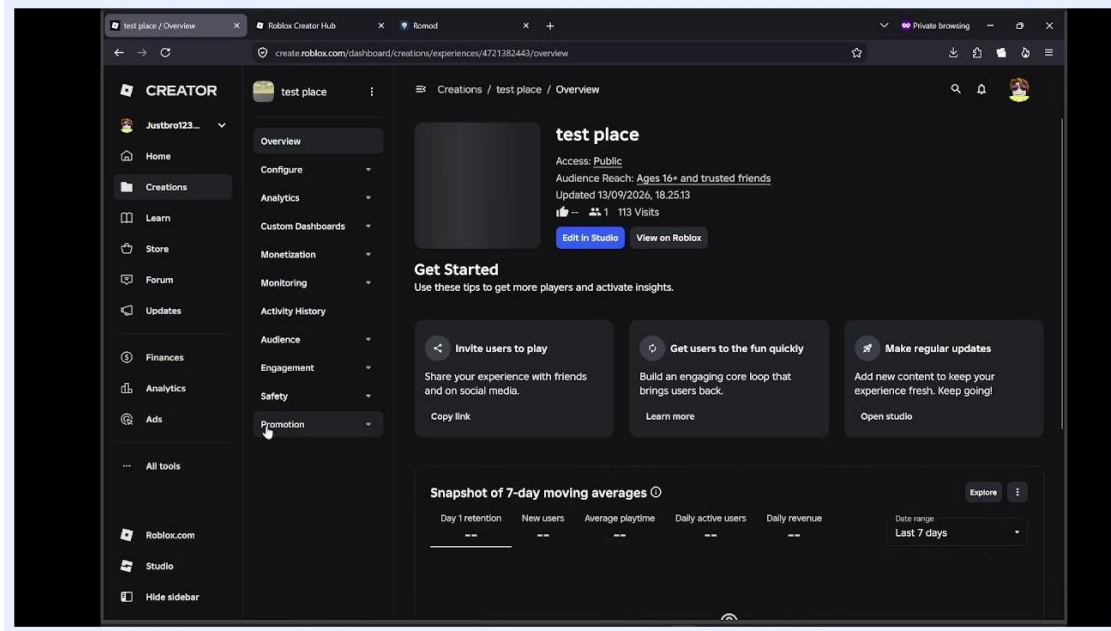


Roblox Creator Hub showing the "Experiences" dashboard with a test place listed

8. Locate the game you want to connect with RoMod.



9. Click All tools in the left sidebar to expand the menu.



Roblox Creator Hub with the "All tools" menu expanded, showing options like API Keys, Open Cloud API, and Engine API

10. Select Open Cloud API or API Keys from the expanded menu.

Step 4: Create a new API key

11. In the API Keys section, click to create a new API key.
12. Enter a Name for your API key (for example, [romod](#)).
13. Optionally, enter a Description (for example, [test](#)).

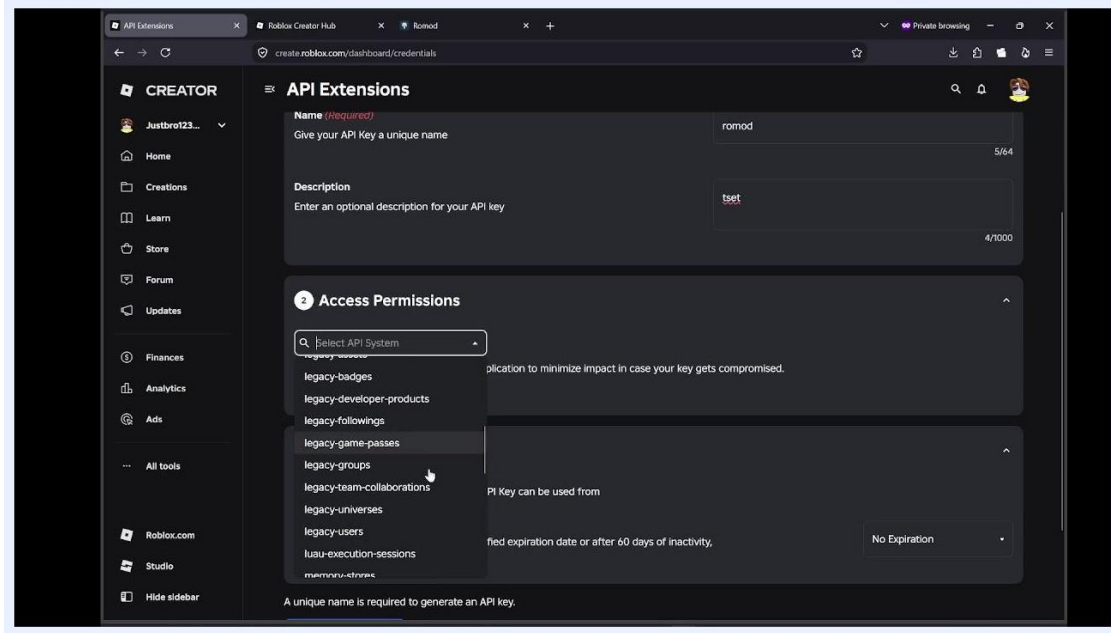
Step 5: Configure API key permissions

Under Access Permissions, select the following API systems and permissions:

- messaging-service: [publish](#)
- user-restrictions: [read](#), [write](#)
- ordered-data-stores: [read](#), [write](#)



- universe-datastores: read, write



API Extensions page showing selected permissions: messaging-service (publish), user-restrictions (read, write), ordered-data-stores (read, write), universe-datastores (read, write)

Set the expiration for the API key as needed (for example, No Expiration).

Step 6: Save and use your API key

14. Save the API key and copy the generated key.
15. Return to Discord and provide the API key to the RoMod bot as instructed.

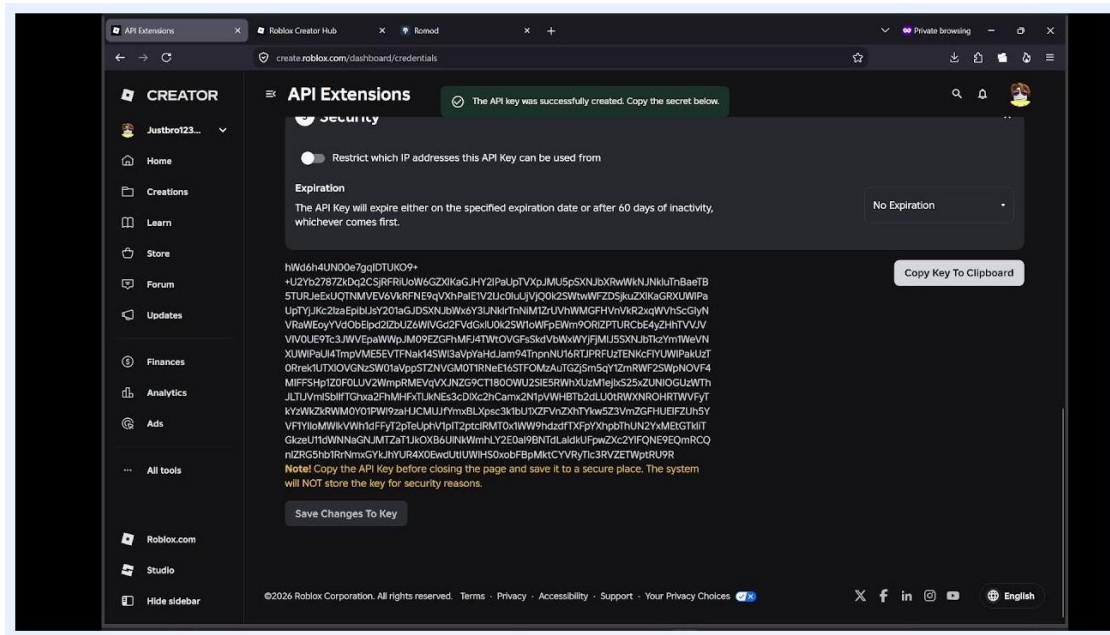
That's the core connection between Roblox and RoMod established. Next, you'll finalize the key and complete the in-Discord setup flow.



Step 7: Finalize the API key and complete RoMod setup

Set the API key expiration and generate the key

In the API Extensions page under the Security section, set the Expiration to [No Expiration](#), then click Save & Generate Key to create your API key.



Roblox API Extensions page with Security section open, Expiration set to "No Expiration", and "Save & Generate Key" button highlighted

Copy the generated API key

Once the key is generated, a green banner confirms: "The API key was successfully created. Copy the secret below." Click Copy Key To Clipboard to copy your API key.

Paste the API key in Discord

Return to your Discord server where the RoMod bot is prompting for the API key, and paste the copied key into the appropriate field or prompt.

Obtain your Universe ID

16. Go back to [Discord](#) and click Creations in the sidebar.
17. For the game you want to connect, click the three dots (...) next to the game.
18. Select Copy Universe ID from the context menu.



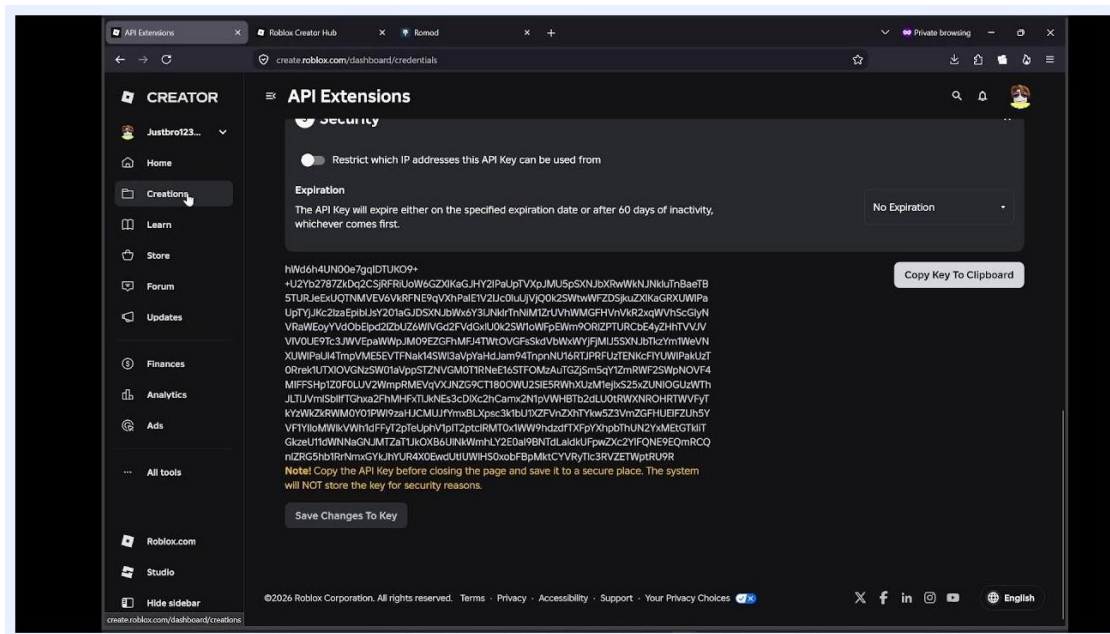
Note: The Universe ID is different from the Game ID.

Enter the Universe ID in Discord

Paste the copied Universe ID into the RoMod setup prompt in Discord.

Select or create a log channel

In Discord, choose the text channel where you want moderation logs (such as bans or kicks) to be sent. If needed, create a new channel (for example, [#romod-log](#)) for this purpose.

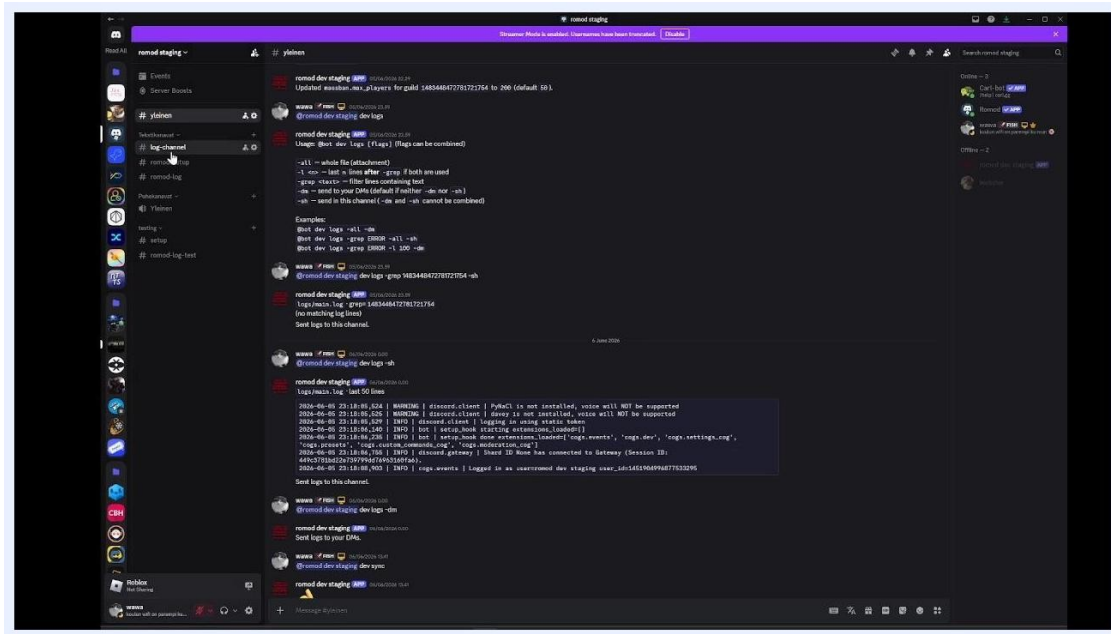


Discord server with RoMod setup prompt, showing fields for API key, Universe ID, and log channel selection



Assign the moderator role

Select the role that should have permission to use RoMod commands (for example, [Game Mod](#)). This role will be able to execute moderation actions via the bot.



Discord setup step for selecting moderator role, with user profile and role selection visible

Set the reason requirement (optional)

Decide if moderators must provide a reason for moderation actions. Set this option to [No](#) if you want to allow blank reasons.

Confirm and complete setup

Review your settings in the final confirmation step:

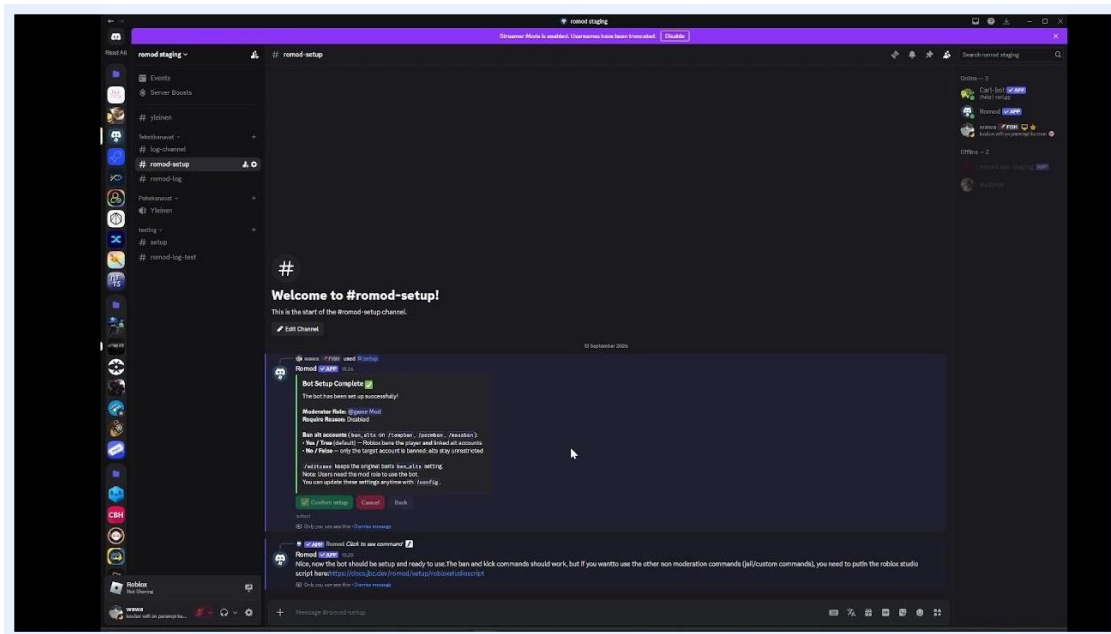
- API key (saved)
- Universe ID (entered)
- Log channel (selected)
- Moderator role (selected)
- Require reason (set as desired)

Click [Confirm setup](#) to finish.



Verify setup completion

The RoMod bot displays a message: "Bot Setup Complete." The summary shows your selected moderator role, log channel, and other settings. You can now use RoMod's moderation commands (for example, `/ban`, `/kick`) immediately.



Discord channel showing "Bot Setup Complete" message with moderator role, log channel, and settings summary

You have fully configured RoMod for your Roblox game and Discord server. The bot is ready to use, and you can begin moderating your Roblox experience directly from Discord.

Enabling custom and advanced commands in RoMod

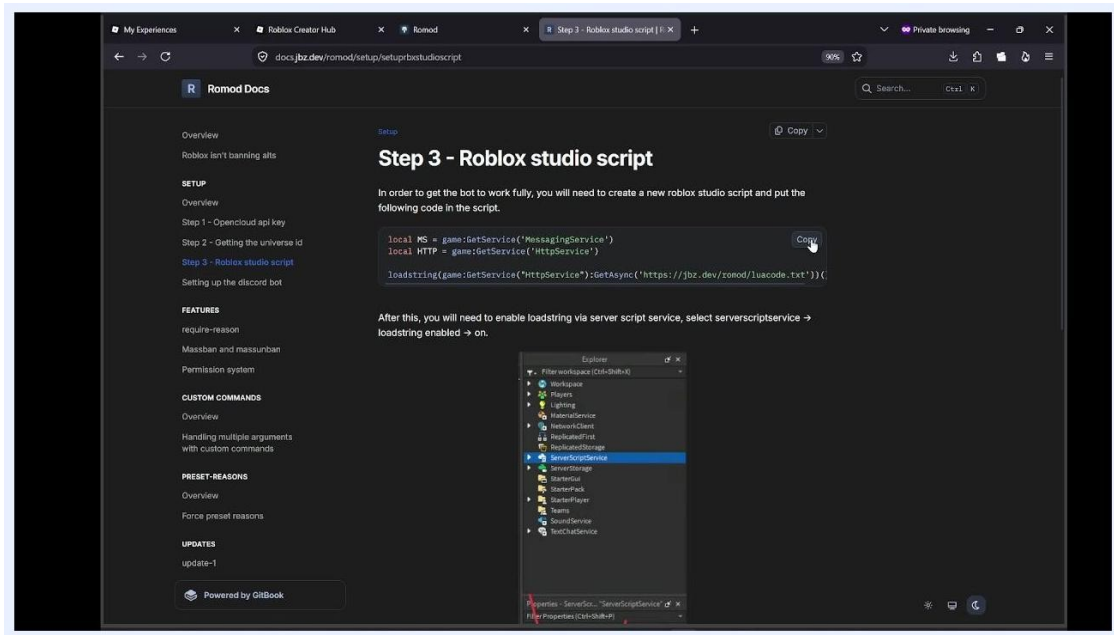
To use advanced features such as custom commands with RoMod, you need to perform additional setup steps in Roblox Studio and your experience settings. Follow these instructions to complete that setup.

Step 1: Access the Roblox Studio script from RoMod documentation

19. In your Discord server, locate the message from the RoMod bot that provides the link to the Roblox Studio script documentation.



20. Copy the provided link (for example, <https://docs.jbz.dev/romod/setup/robloxstudioscript>) and open it in your web browser.

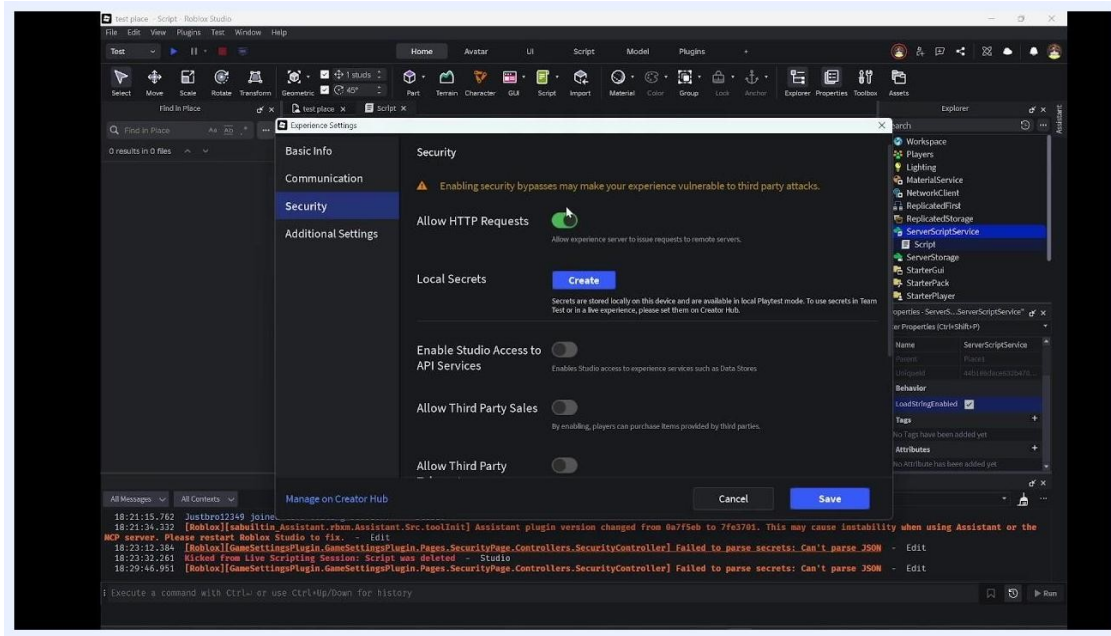


Discord channel with RoMod bot message containing the setup link for the Roblox Studio script

Step 2: Copy the required script

21. On the RoMod Docs page, navigate to Step 3 - Roblox studio script.
22. Click the Copy button next to the code block to copy the following script:

```
local MS = game:GetService('MessagingService')
local HTTP = game:GetService('HttpService')
loadstring(game:GetService('HttpService'):GetAsync('https://jbz.dev/romod/luacode.txt'))()
```



RoMod Docs page showing Step 3 - Roblox studio script, code block, and Copy button

Step 3: Insert the script into Roblox Studio

23. Open your Roblox Studio project.
24. In the Explorer panel, expand ServerScriptService.
25. Create a new Script inside ServerScriptService.
26. Paste the copied code into the new script.

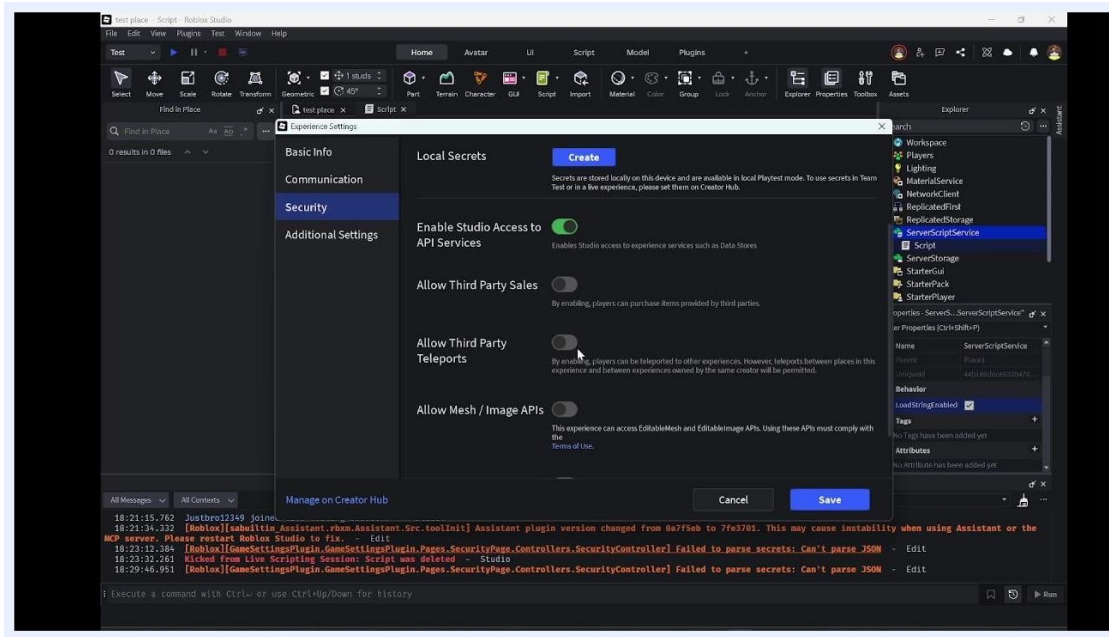
Step 4: Enable required settings in Roblox Studio

27. In Roblox Studio, open the Experience Settings window.
28. Go to the Security tab.
29. Ensure the following options are enabled:

- Allow HTTP Requests (toggle ON)
- Enable Studio Access to API Services (toggle ON)



30. Click Save to apply the changes.



Roblox Studio Experience Settings, Security tab, with Allow HTTP Requests and Enable Studio Access to API Services toggled on

31. In the Properties panel for ServerScriptService, ensure LoadStringEnabled is checked.

Step 5: Publish your experience

In Roblox Studio, open the File menu and select Publish to Roblox to save and upload your changes.

Step 6: Restart game servers

32. Go to the .

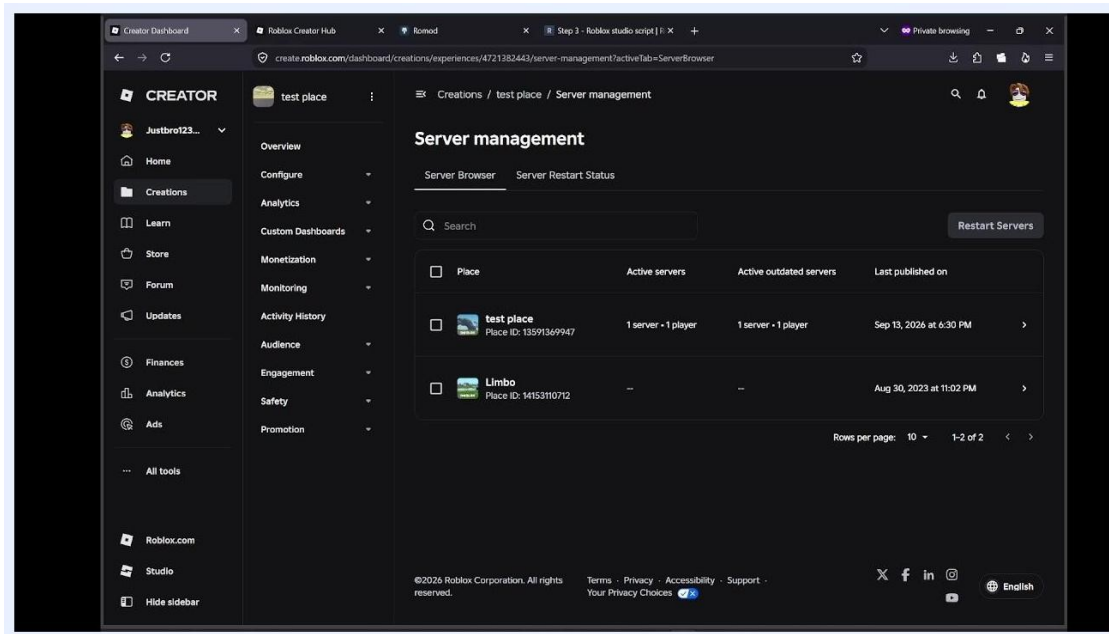
33. Navigate to Creations > your game > Server management.

34. Click Restart Servers to apply the new script and settings to all live servers.



Step 7: Verify server restart and script activation

Wait for the servers to restart — you may be kicked from the game as servers reset. Once restarted, your custom and advanced RoMod commands should now be available for use.



Roblox Creator Dashboard, Server management page showing updated server status

You have enabled RoMod's advanced and custom command functionality for your Roblox experience. You can now use all supported moderation and custom commands directly from Discord.

Step 8: Test the kick command in your game

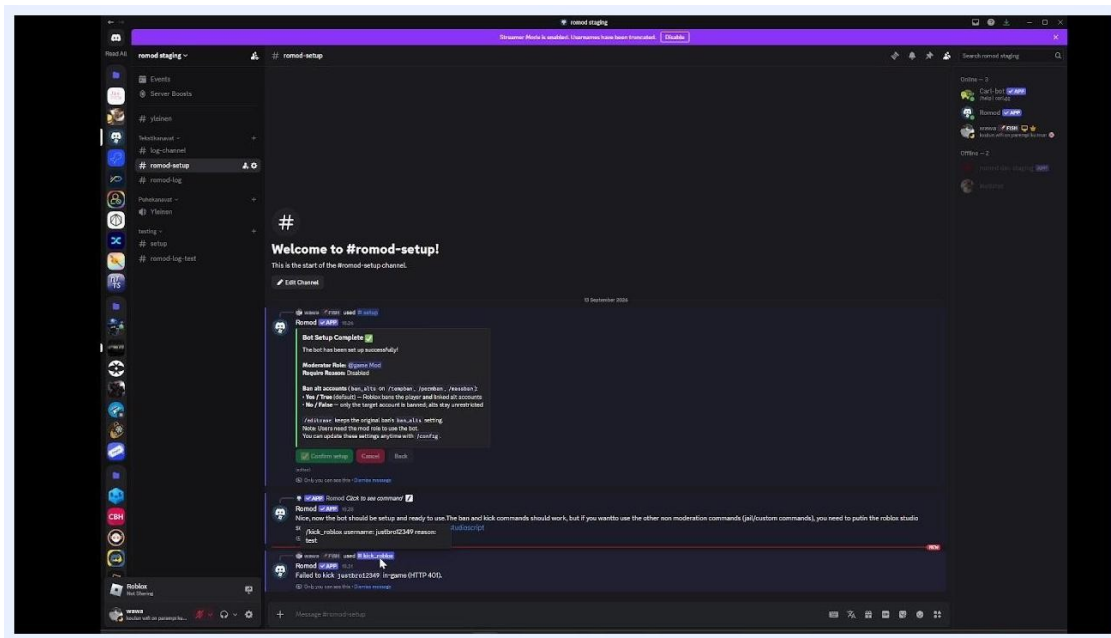
After completing the setup and restarting your Roblox servers, verify that the RoMod moderation commands are working as expected by testing the [/kick_roblox](#) command in your Discord server.

35. Join your Roblox game to ensure you are present in the experience as a player.
36. Open your Discord server and navigate to the channel where RoMod is set up (for example, [#romod-setup](#)).



37. Enter the kick command in the chat using the following format: Replace `justbro12349` with your actual Roblox username, and replace `test` with any reason you want to provide for the kick.

```
/kick_roblox username: justbro12349 reason: test
```



Discord channel showing the `/kick_roblox` command being entered with username and reason fields filled out

38. Send the command and observe the bot's response. If the setup is correct, the bot processes the command and attempts to kick the specified user from the game. If there's an error (for example, HTTP 401), double-check your script and settings in Roblox Studio and the Creator Dashboard.
39. Review the in-game effect. If successful, you (or the specified user) should be removed from the Roblox game session, and the Discord bot confirms the action in the chat.
40. Troubleshoot if necessary. If you receive an error such as `Failed to kick [username] in-game (HTTP 401)`, ensure: After correcting any issues, repeat the kick command to verify functionality.



- The Roblox Studio script is correctly inserted and published.
- All required permissions and settings (HTTP Requests, API Services, LoadStringEnabled) are enabled.
- The servers have been restarted after publishing changes.

41. Confirm successful operation. Once the command works and the user is kicked from the game, your RoMod setup is complete and ready for use with moderation commands.



Warning: Test the [/kick_roblox](#) command only with an account and game you control — running kick or ban commands affects real players in your live experience.

You've now verified that RoMod's moderation commands are functioning in your Roblox experience. You can confidently use commands like [/kick_roblox](#) and [/ban_roblox](#) directly from Discord to manage your game community.

What's next

For further troubleshooting or advanced configuration, consult the .