

# Advanced Sneaks Grants Extractor - Installation Guide

---

## 1. Overview

---

This document provides comprehensive instructions for the installation and operation of the **Advanced Sneaks Grants Extractor**. This sophisticated Python-based utility is engineered to retrieve `grants.txt` (specifically, the `grants_token`) from FiveM servers. It features a modern, interactive command-line interface, designed for users requiring server grants for resources not officially owned on Keymaster, facilitating integration with the 'Connect' system of our Discord bot for asset decryption.

**Critical Notice:** This tool is strictly intended for use in conjunction with our Discord bot. Independent use of extracted grants or operation outside the integrated bot environment is not supported and will result in operational failure. For support and decryption services, please join our Discord community: <https://discord.gg/yRak5SEp4H>.

## 2. Key Features

---

- **Modern User Interface:** Utilizes the `rich` library for a sleek, grey-themed terminal experience, incorporating progress indicators, structured panels, and formatted tables.
- **Operational Warning System:** Displays a prominent warning panel at startup, delineating the tool's intended use-case and inherent limitations.
- **Flexible Target Input:** Accommodates both direct server IP addresses (e.g., `127.0.0.1:30120`) and FiveM CFX join links (e.g., `cfx.re/join/xxxxxx`).
- **Optional CitizenFX Token:** Supports the inclusion of an `X-CitizenFX-Token` header for servers mandating authentication.
- **Active Connection Requirement:** Mandates an active connection to the target FiveM server prior to grant extraction. The tool will verify this prerequisite.

- **Automatic Grant Persistence:** Automatically saves the retrieved `grants_token` to a user-specified output file (default: `grants.txt`).
- **Debug Mode:** An optional `--debug` flag provides verbose output, invaluable for advanced troubleshooting and diagnostic purposes.

## 3. Setup and Prerequisites

---

Successful deployment of the Advanced Sneaks Grants Extractor necessitates the presence of Python 3 on your system, alongside specific external Python libraries.

### 3.1. Python Installation

Ensure **Python 3.11** or a later version is installed on your operating system.

[Download Python](#)

### 3.2. Dependency Installation

Execute the following command in your terminal or command prompt to install the requisite Python libraries (`rich` and `requests`):

```
pip install rich requests
```

## 4. Usage Protocol

---

Prior to initiating the Advanced Sneaks Grants Extractor, it is imperative that you are **actively connected to the target FiveM server** within your game client. The tool will perform a stringent validation of this active connection; failure to meet this requirement will result in process termination.

### 4.1. Script Preparation

1. Download the `main.py` script.
2. Place the script in a designated directory on your system.

## 4.2. Execution Syntax

Navigate to the directory containing `main.py` in your terminal and execute the tool using the following command structure:

```
python main.py <TARGET>
```

Replace `<TARGET>` with either the server's IP address and port (e.g., `127.0.0.1:30120`) or a CFX join link (e.g., `cfx.re/join/xxxxxx`).

### Example with Server IP:

```
python main.py 127.0.0.1:30120
```

### Example with CFX Join Link:

```
python main.py cfx.re/join/xxxxxx
```

## 4.3. Interactive Mode

Should the script be executed without a `<TARGET>` argument, it will automatically engage an interactive mode, prompting for the necessary server information:

```
python main.py
```

Adhere to the on-screen prompts to furnish the required details.

## 4.4. Command-Line Arguments

The tool supports the following optional command-line arguments:

Argument	Description	Default Value
<TARGET>	The target server IP:PORT or CFX Join Link.	(Interactive)
--token	Optional CitizenFX Token for authenticated access.	"" (empty string)
--output	The filename for saving the extracted grants.	grants.txt
--debug	Activates detailed debug logging for diagnostics.	False

### Example with Token, Custom Output File, and Debug Mode:

```
python main.py 127.0.0.1:30120 --token YOUR_CFX_TOKEN --output  
my_server_grants.txt --debug
```

## 5. Output Interpretation

---

Upon successful grant extraction, the tool will present a concise summary of the operation, detailing the resolved host, connection status, token validation status, and a truncated hash of the extracted grants token. The complete `grants_token` will be securely saved to the designated output file.

In instances of operational failure, the tool will issue a specific warning message, elucidating the cause of the failure (e.g., connection anomalies, server-side errors, or the absence of a `grants_token` in the server's configuration).

## 6. Troubleshooting Guide

---

- **“Connection Failed”**: Verify the accuracy of the server IP and port. Confirm the server's online status and network accessibility. Investigate any potential firewall restrictions.
- **“No Active Connection Detected”**: This critical error signifies the tool's inability to confirm an active connection to the target FiveM server. **An active connection to the server is mandatory** for the tool's operation. Ensure your FiveM client is running and successfully connected to the specified server.

- **“Server Rejected Request” or “Token Not Exposed by Server”**: This typically indicates an unexpected response from the server’s `/client` endpoint or the absence of the `grants_token`. This can stem from server configuration, network impediments, or active request blocking. Utilize the `--debug` flag for verbose diagnostic output.
- **“Failed to Resolve CFX Code”**: If a CFX join link was provided and this error occurred, the tool was unable to resolve the corresponding server endpoint. In such scenarios, procure the direct IP:PORT of the server and employ it as the target.
- **“File Write Error”**: Ascertain that the script possesses the necessary file permissions within its execution directory to create or modify the output file.

## 7. Disclaimer

---

This tool is furnished solely for informational and specialized utility purposes. The developers disclaim all responsibility for any misuse or adverse consequences arising from the deployment of this software. Users are unequivocally advised to adhere to all applicable legal statutes and terms of service. Always secure the requisite permissions prior to attempting to access or extract data from any server.

## 8. References

---

[1] Python Official Website. (n.d.). *Python Downloads*. Retrieved from <https://www.python.org/downloads/> [2] Requests: HTTP for Humans. (n.d.). *Requests Documentation*. Retrieved from <https://requests.readthedocs.io/en/master/> [3] Rich: A Python library for rich text and beautiful formatting in the terminal. (n.d.). *Rich Documentation*. Retrieved from <https://rich.readthedocs.io/en/stable/>

---